



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

**IMPLEMENTACE TVAROVÁNÍ ANTÉNNÍCH PŘÍJMOVÝCH
SVAZKŮ RADARU V FPGA**

RADAR RECEIVER BEAMFORMING IMPLEMENTATION IN FPGA

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Bárta

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Marek Bohrn, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**
Ústav mikroelektroniky

Student: Bc. Jakub Bárta

ID: 158102

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Implementace tvarování anténních příjmových svazků radaru v FPGA

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a implementujte algoritmus příjmu dat jednotlivých datových streamů z anténních řad. Navrhněte způsob synchronizace jednotlivých datových streamů. Navrhněte a implementujte algoritmus tvarování (výpočtu vzorků) jednotlivých svazků. Vyhodnoťte důsledky výpadku některého z datových toků (dat některé z řad antény). Navrhněte a implementujte algoritmus odesílání výsledných dat. Návrh proveďte konfigurovatelný za provozu (počet výstupních svazků, koeficienty rozložení amplitud, rozsah zpracovávaných vzorků). Návrh proveďte pro FPGA Altera (Intel) v jazyce VHDL.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 4.2.2019

Termín odevzdání: 19.8.2019

Vedoucí práce: Ing. Marek Bohrn, Ph.D.

Konzultant:

doc. Ing. Lukáš Fucík, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá návrhem, implementací a ověřením funkce digitálního tvarovače příjmových svazků 3D radaru. V textu práce je odvozen algoritmus tvarování a detailně popsána jeho implementace na desce osazené obvodem Cyclone V. V závěru je provedeno ověření funkce tvarovače a popsány možnosti jeho dalšího využití.

KLÍČOVÁ SLOVA

3D radar, fázovaná anténní řada, vyzařovací charakteristika, AESA, PESA, Butlerova matice, tvarování svazku, vychylování svazku, s-parametry, monopolzní detekce, FPGA, Altera, Cyclone V, SoC,

ABSTRACT

This thesis deals with design and implementation of digital beamformer for 3D radar. The text of this thesis contains derivation of beamforming algorithm and detailed description of it's implementation on development kit with Cyclone V circuit. At the end of the thesis the beamformer design is verified and it's further usage is discussed.

KEYWORDS

3D radar, phased antenna array, antenna pattern, AESA, PESA, Butler matrix, beamforming, beamsteering, monopulse detection, s-parameters, FPGA, Altera, Cyclone V, SoC

BÁRTA, Jakub. *Implementace tvarování anténních příjmových svazků radaru v FPGA*. Brno, 2019, 58 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce: Ing. Marek Bohrn, Ph.D.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Marku Bohrnovi, Ph.D. za trpělivost a konzultace k práci. Rovněž bych rád poděkoval firmě Retia a.s. jmenovitě p. Davidu Párovi za zadání práce, poskytnutí vybavení a hodnotných informací k vypracování této práce. Další díky patří p. Šubrtovi za pomoc při řešení problémů s implementací a p. Šedivému za poskytnutí obrazového materiálu a teoretických znalostí nezbytných k vypracování textu práce.

PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Obsah

Úvod	9
1 Anténní řady	10
1.1 Vyzařovací charakteristika	11
1.1.1 Vychylování	12
1.1.2 Tvarování	13
1.2 Způsoby tvarování svazků	18
1.2.1 Butlerova matice	18
1.2.2 PESA	19
1.2.3 AESA	19
1.3 Monopulzní určení polohy cíle	22
1.4 Vliv reálné signálové cesty	24
2 Integrace tvarovače do radaru	26
2.1 Vývojová deska	27
2.1.1 FPGA Cyclone V	28
3 Návrh tvarovače	30
3.1 Realizovaná funkce	30
3.2 Způsob implementace	31
3.2.1 Výpočetní blok	32
3.2.2 Paměť FIFO	37
3.2.3 Propojení IP Jader	38
3.2.4 Program pro HPS	39
3.2.5 Podpůrné programy	42
4 Ověření funkce	45
4.1 MATLAB model	45
4.2 Porovnání tvarovače s modelem	49
4.3 Vyhodnocení	51
4.4 Využití zdrojů obvodu FPGA	51
4.5 Zvyšování přesnosti	52
4.6 Další rozšíření	54
Závěr	55
Literatura	56
Seznam symbolů, veličin a zkratk	58

Seznam obrázků

1.1	Princip anténní řady	10
1.2	Dráhový rozdíl při dopadu signálu pod úhlem	12
1.3	Pascalův trojúhelník	13
1.4	Porovnání rozložení amplitud	15
1.5	Anténní charakteristika bez funkce okna (obdélník)	15
1.6	Anténní charakteristika s Binomickým polem	16
1.7	Anténní charakteristika s Čebyševovým oknem	16
1.8	Anténní charakteristika s Taylorovým oknem	17
1.9	Zapojení Butlerovy matice	18
1.10	Blokové schéma PESA	19
1.11	Blokové schéma AESA	20
1.12	Anténa s elektronicky řízeným svazkem v jedné ose	20
1.13	Konfigurace antény s řízením v obou osách	21
1.14	Frekvenčně závislá anténa	22
1.15	Monopulzní detekce cíle	23
1.16	Vyhodnocení úhlové odchylky	23
2.1	Blokové schéma demonstrátoru	26
2.2	Vývojový kit DE0-nano	27
2.3	Blokové schéma HPS	28
3.1	Blokové schéma tvarovače	32
3.2	Blokové schéma výpočtu 1 kanálu	33
3.3	Časový průběh komunikace na sběrnici Avalon Stream	36
3.4	Realizace ovládacích signálů sběrnice Avalon stream	36
3.5	Nastavení parametrů IP bloku paměti FIFO	38
3.6	Blokové schéma projektu	39
3.7	Vývojový diagram serverové aplikace	40
4.1	Součtový signál svazku při nulovém vychýlení od osy antény	46
4.2	Difrakční lalok při vychýlení 10°	47
4.3	Difrakční lalok při vychýlení 30°	48
4.4	Průběh součtového signálu při vychylování svazku	49
4.5	Demonstrátor 3D radaru na střeše firmy Retia a.s.	50
4.6	Histogram odchylek vzniklých zaokrouhlováním	51
4.7	Odchylky výsledků po rozšíření koeficientů	53

Seznam tabulek

2.1	Dostupné zdroje obvodu 5CSEMA4U23C6N [21]	29
3.1	Struktura přijímaného paketu [22]	41
3.2	Struktura souboru koeficientů	43
4.1	Využití zdrojů obvodu	52
4.2	Statická časová analýza	52

Úvod

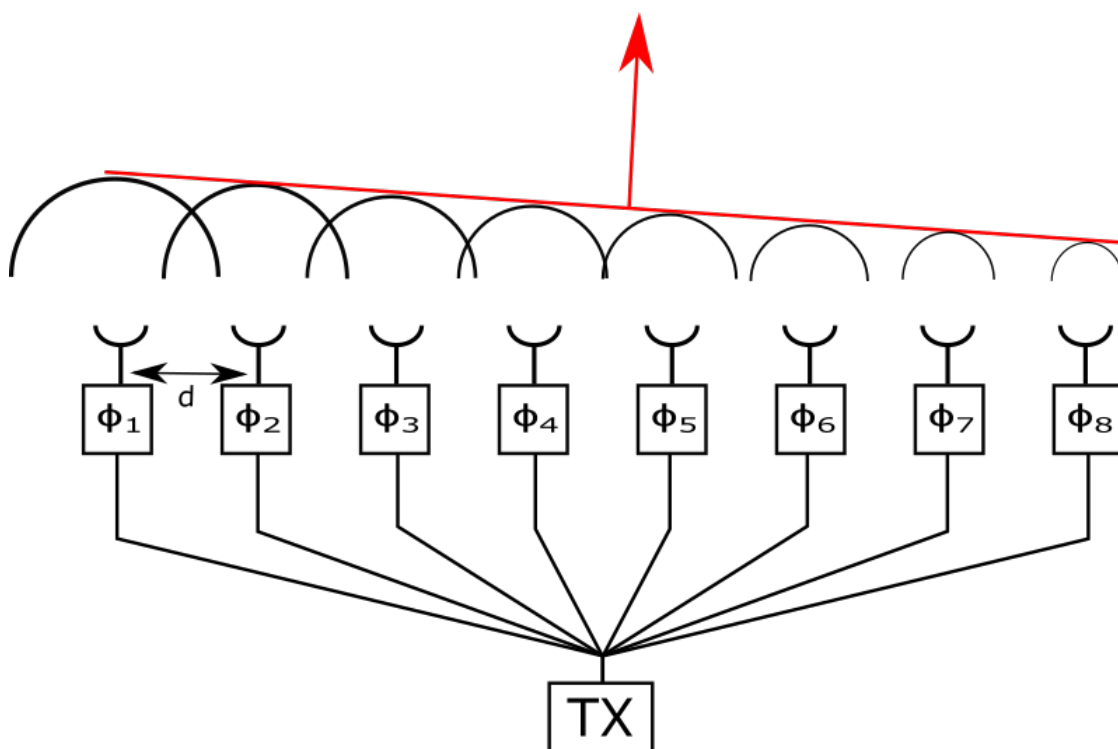
Předmětem této práce je návrh algoritmu tvarovače příjmových anténních svazků a jeho implementace do obvodu FPGA. Tvarovač je určen pro 3D radar typu AESA s digitalizací signálu jednotlivých řad. Díky využití paralelního zpracování dat v FPGA by měl být schopen poskytnout větší výpočetní výkon, než tvarovače realizované signálovými procesory.

Klíčovou vlastností navržené implementace musí být snadná modifikace pro možnost změny režimů funkce. Dále by měla být snadno přenositelná na jiná zařízení a obvody FPGA firmy Intel.

Práce v kapitole „anténní řady“ obsahuje teoretický popis implementovaných funkcí, které vstupují do tvarování příjmových anténních svazků a objasňuje důvody jejich využití. Poté je v kapitole „Integrace do radaru“ popsáno zařazení implementovaného tvarovače v rámci struktury 3D radaru a popsán vývojový kit, na kterém bude tvarovač testován. Kapitola „návrh tvarovače“ detailně popisuje implementaci všech částí návrhu stejně jako dalších programů nezbytných pro jeho chod. V kapitole „ověření funkce“ poté následuje popis scénáře testování navrženého tvarovače a výsledky ověření jeho funkce.

1 Anténní řady

Navrhovaný tvarovač svazků má za úkol zpracovávat signál z anténní řady. Pojmem fázovaná anténní řada se rozumí anténa složená z více vysílacích prvků na které může být signál aplikován s různým fázovým zpožděním (obrázek 1.1) [1][2][4][6]. Fázovaná anténní řada využívá ke své funkci interference. V místech, kde jsou vyslané signály ve fázi, dochází k jejich zesilování. Naopak v místech, kde je fáze signálů opačná, se zeslabují.



Obr. 1.1: Princip anténní řady

Výsledkem je anténa s vysokým ziskem, možností potlačení postranních laloků a odolností vůči poruše jednotlivých prvků. Nevýhodou je úzká vazba na pracovní frekvenci, která je dána vzdálenostmi prvků antény. Fázované antény bývají využívány především kvůli možnosti elektronicky řídit jejich svazek. V radarových aplikacích se nejčastěji využívá plošná řada, která má prvky rozmístěné do matice ve 2 osách.

1.1 Vyzařovací charakteristika

Vyzařovací charakteristika maticové antény závisí na vlastnostech jednotlivých prvků, jejich počtu a rozmístění [1],[10]. Při návrhu antény je nutné nejprve stanovit požadované pokrytí prostoru, pracovní frekvenci a zisk antény. Od toho se odvíjí konstrukce anténní řady.

U jednotlivých prvků předpokládáme obecně stejnou směrovou charakteristiku. Prvky mohou být tvořeny dipóly, trychtýřovými anténami, ale bez problému i dalšími řadami. Příkladem může být Yagi anténa. Prvky je možné rozložit nejen v rovině, ale též po zakřivené ploše.

Vzdálenost prvků je odvozena od frekvence (resp. vlnové délky) vysílaného/přijímaného signálu. Používaný rozsah vzdáleností prvků je od desetin až po jednotky vlnové délky. Menší vzdálenosti způsobí ztrátu směrových vlastností antény a její vyzařovací charakteristika bude stejná, jako u jednoho prvku. Větší vzdálenost bude zužovat hlavní lalok, ale vytvoří významné postranní laloky jejichž počet bude se vzdáleností prvků narůstat.

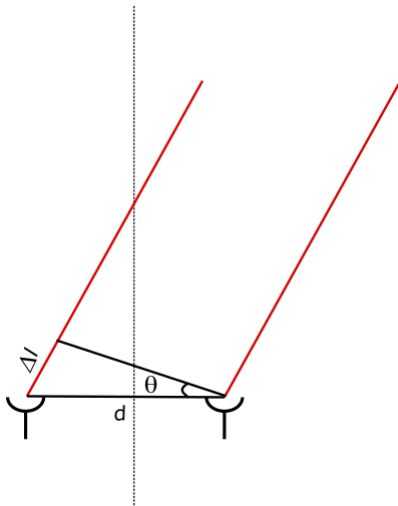
V praxi se volí vzdálenost nejčastěji mezi $0,5$ a $0,8\lambda$. U spodní hranice je hlavní lalok široký, ale postranní laloky nevznikají. Při vzdálenosti větší než $0,5\lambda$ se hlavní laloku zužuje. Při vychylování svazku takovéto antény navíc dochází ke vzniku difrakčních postranních laloků o stejné velikosti, jako hlavní lalok [4].

Dalším faktorem ovlivňujícím vyzařovací charakteristiku je počet prvků soustavy. Se zvyšujícím se počtem stoupá celkový zisk antény a hlavního laloku, ale zároveň se přidávají postranní laloky s nižším ziskem. Počet prvků se v praxi volí co největší. Limitujícím faktorem je v tomto případě velikost antény a cena.

U jednotlivých elementů antény mohou být nezávisle řízeny amplituda a fáze. Změna amplitudy je využívána ke tvarování svazků, především zmenšování postranních laloků [7]. Řazení fázového zpoždění je využíváno k vychylování svazku [8]. Pro zvýšení přesnosti vyhodnocování azimutu a elevace je možné dále obě tyto techniky kombinovat s monopolní detekcí cíle.

1.1.1 Vychylování

Pokud na anténní řadu dopadá signál pod určitým úhlem (obr.1.2), bude signál zachycený různými elementy řady fázově (časově) posunutý. Tento jev je způsoben rozdílem vzdálenosti Δl , kterou musí vlna překonat.



Obr. 1.2: Dráhový rozdíl při dopadu signálu pod úhlem

Časový posun Δl je dán vzdáleností prvků antény d a úhlem dopadu vlny θ (rovnice 1.1).

$$\Delta l = d \cdot \sin \theta \quad (1.1)$$

Po dosazení zpoždění do rovnice vlny a úpravě bude pro n -tý prvek antény platit rovnice 1.2.

$$y_n(t) = x_n(t) \cdot \sin(2\pi f t + (n - 1) \cdot \frac{2\pi}{\lambda} \cdot d \cdot \sin \theta), \quad (1.2)$$

kde y_n je výstup n -tého prvku řady v čase t , x je přijatý signál, f je nosná frekvence, λ příslušná vlnová délka a θ úhel dopadu signálu. Pro celou řadu je výstup dán rovnicí 1.3.

$$Y(t) = \sum_1^N X \cdot e^{j2\pi(n-1)\frac{d}{\lambda}\sin(\theta)} \quad (1.3)$$

Exponenciální část rovnice představující přenos antény bývá označována také jako komplexní tvarovací váha (AF) [8]. Takto získaná funkce lze použít i obráceně a dosazením požadovaného úhlu vychýlit svazek antény. Při aplikaci tvarovacích vah na přijímaný signál lze takto získat více svazků z jedné periody.

1.1.2 Tvarování

Na všechny elementy řady není nutné nahlížet jako na rovnocenné. Vhodným rozdělením významu elementů řady za pomoci koeficientů mohou být zmenšeny nebo dokonce zcela odstraněny postranní laloky. Výjimkou jsou difrakční laloky, které lze odstranit pouze změnou rozestupu prvků antény. Nevýhodou změny váhování prvků naopak může být rozšíření hlavního laloku (zhoršení směrovosti) a snížení celkového zisku antény (zmenšení odstupu signál-šum) [1],[7],[10]. Pro výpočet koeficientů řady je možné volit různé okenní funkce. V této kapitole bude představeno a porovnáno několik nejběžnějších z nich.

Binomické pole

Prvním z možných rozdělení amplitud je využití koeficientů Pascalova trojúhelníku (Obr:1.3).

				1					
			1		1				
		1		2		1			
	1		3		3		1		
1		4		6		4		1	
1	5	10	10	5	1				
1	6	15	20	15	6	1			

Obr. 1.3: Pascalův trojúhelník

Tato metoda je nejvýhodnější z hlediska odstranění postranních laloků, neboť je odstraní úplně. Nevýhodou je omezená možnost aplikace vzhledem ke značnému rozptylu koeficientů, který znesnadňuje výpočty.

Dolph-Čebyševovo okno

Pro dosažení lepší směrovosti a menšího rozptylu hodnot se často využívá okno Dolph Čebyševovy funkce. Ta oproti Pascalovu trojúhelníku neodstraní postranní laloky kompletně, ale pouze omezí na stejnou úroveň. Rozdělení popisují rovnice 1.4, 1.5, kde M je počet prvků antény a A požadované potlačení postranních laloků v dB.

$$w(\omega_k) = \frac{\cos\{M \cos^{-1}[\beta \cos(\frac{\pi k}{M})]\}}{\cosh[\frac{1}{M} \cosh^{-1}(\beta)]} \quad k = 0, 1 \dots M - 1 \quad (1.4)$$

$$\beta = \cosh \left(\frac{1}{M} \cosh^{-1} (10^{\frac{A}{-20}}) \right) \quad (1.5)$$

Amplitudy funkce směrem od středu řady klesají, na krajích však při velkém počtu prvků dochází k prudkému nárůstu. Přestože se tento výpočet může zdát komplikovaný, je pomocí výpočetní techniky daleko lépe realizovatelný než Pascalův trojúhelník. Hodnoty lze získat například v programu MATLAB funkcí *chebwin(L,r)* obsaženou v *System toolbox*, kde L je velikost pole a r Velikost potlačení postranních laloků [16].

Taylorova funkce

Taylorovo (\bar{n}) rozložení amplitudy vychází z Dolph-Čebyševova a bylo vytvořeno ve snaze získat „ideální rozložení“. Je jakýmsi kompromisem mezi binomickým a Dolph-Čebyševovým oknem. Za cenu menší směrovosti řeší problém s nárůstem amplitudy u okrajů řady. Postranní laloky nejsou konstantní, ale postupně se zmenšují. Tato výhoda se uplatní především v případě dlouhé řady, kdy u Dolph-Čebyševova okna dochází ke zvětšení postranních laloků. Koeficienty jsou vypočteny podle rovnice 1.8. Volenými parametry funkce jsou počet prvků řady, řád Taylorova polynomu a velikost největších postranních laloků.

$$A(n) = 2 \cdot \sum_1^M F_m \cos \left(2\pi m \frac{k_n - \frac{1}{2}N + \frac{1}{2}}{N} \right) \quad (1.6)$$

$$F_m = \frac{-1^{(m+1)} \cdot a_m}{2b_m} \quad (1.7)$$

$$a_m = \prod_{x=1}^M \left(\frac{1 - \left(\frac{m^2}{\tau}\right)}{A^2 + \left(x - \frac{1}{2}\right)^2} \right); \tau = \frac{M^2}{A^2 + \left(M - \frac{1}{2}\right)^2}; A = \frac{\cosh \left(10^{-\frac{SLL}{20}} \right)}{\pi} \quad (1.8)$$

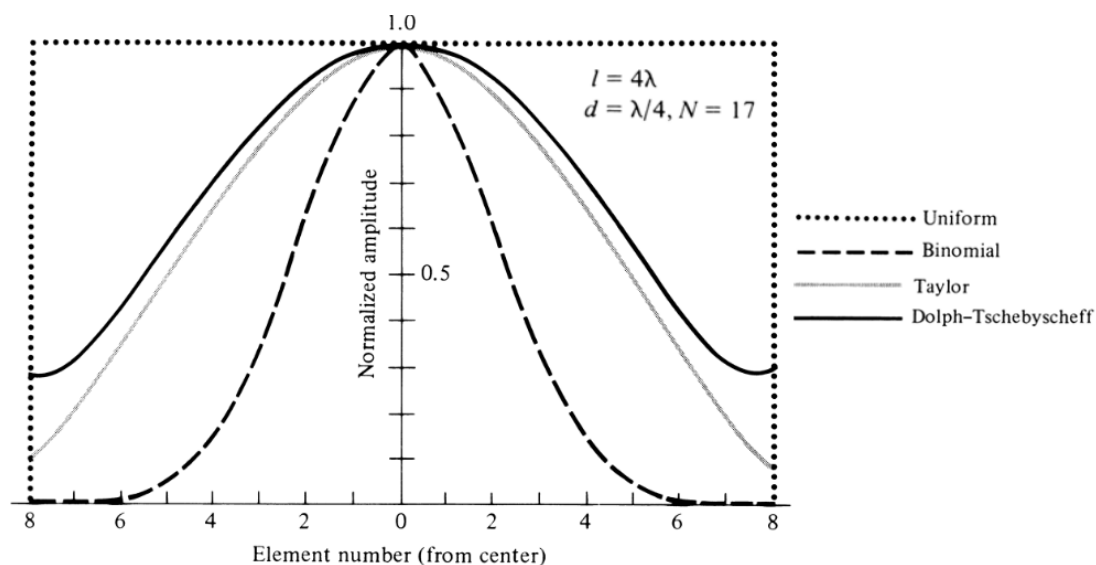
$$b_m = \prod_{y=1}^{M-1} \left(\frac{1 - m^2}{y^2} \right), \quad (1.9)$$

kde: M je řád Taylorova polynomu,
 N je počet prvků řady,
 $k \in \langle 0, N - 1 \rangle$,
 SLL je požadovaná velikost postranních laloků.

Pro praktické účely není stejně jako u Dolph-Čebyševovy funkce nutné výpočet provádět ručně. Pro výpočet se běžně využívá funkce *taylorwin(L,nbar,sll)*, kde L je velikost pole, $nbar$ počet postranních laloků stejné velikosti, sll maximální velikost postranního laloku [15].

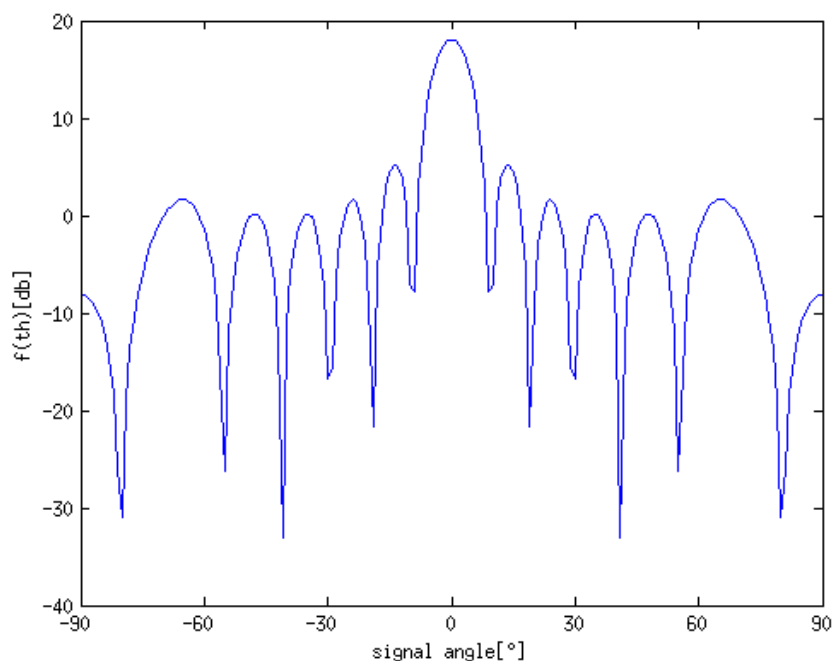
Porovnání

Porovnání průběhů výše zmíněných funkcí je viditelné na obrázku 1.4.

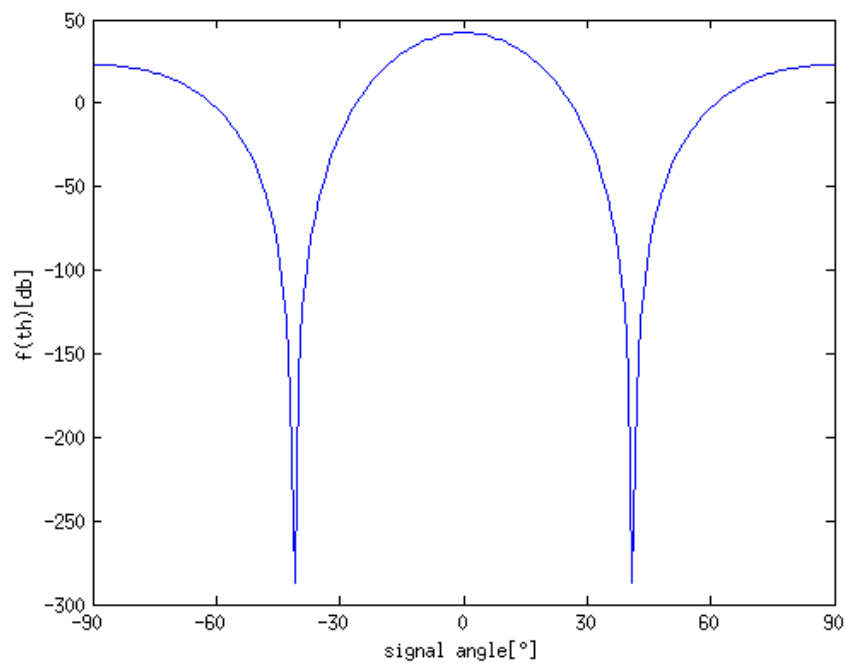


Obr. 1.4: Porovnání rozložení amplitud [7]

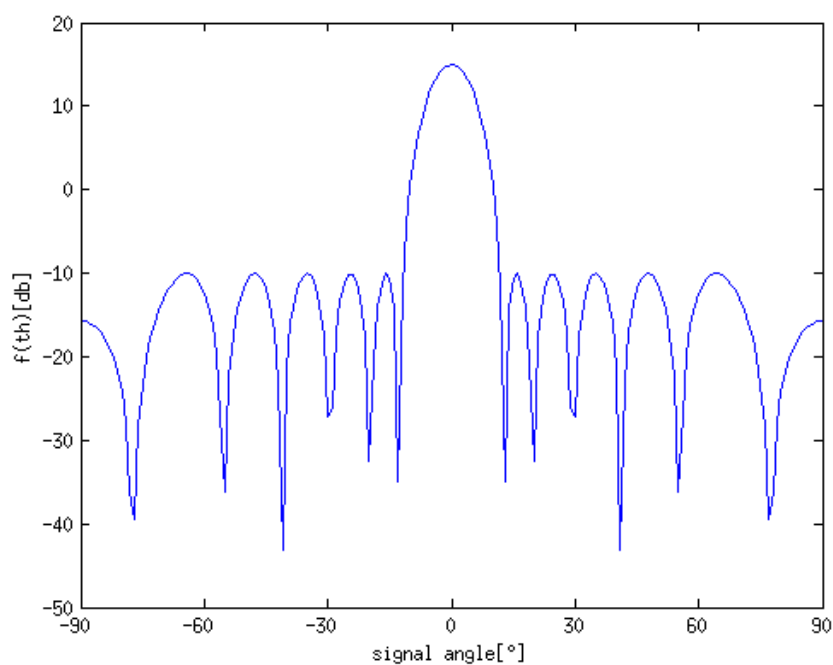
Průběh okenních funkcí je pouze ilustrační a nijak nevypovídá o výsledném vlivu na příjmovou charakteristiku antény. Pro korektní porovnání je nutné funkci aplikovat na anténu a následně zjistit její charakteristiku. Vykreslené křivky na obrázcích jsou získány z modelu antény vytvořeném v programu MATLAB.



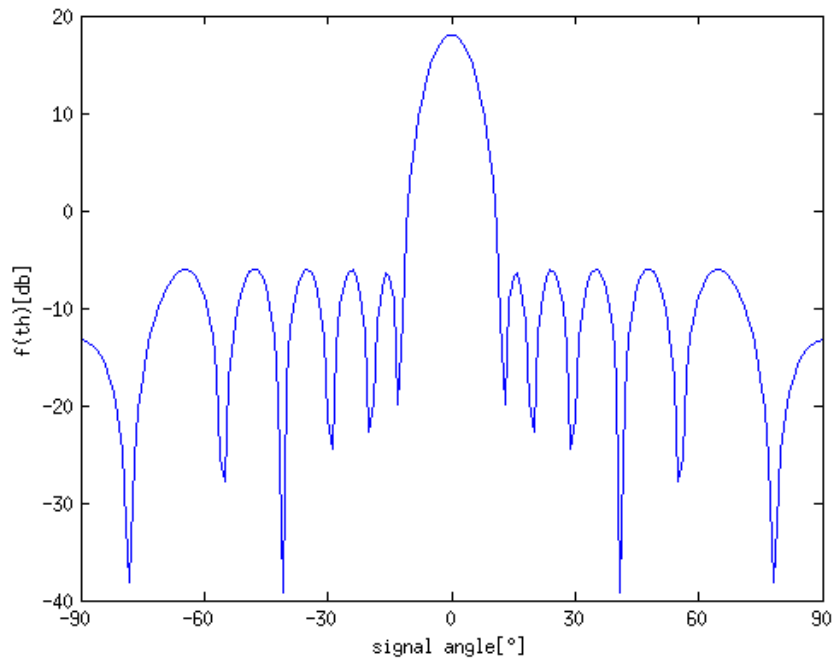
Obr. 1.5: Anténní charakteristika bez funkce okna (obdélník)



Obr. 1.6: Anténní charakteristika s Binomickým polem



Obr. 1.7: Anténní charakteristika s Čebyševovým oknem



Obr. 1.8: Anténní charakteristika s Taylorovým oknem

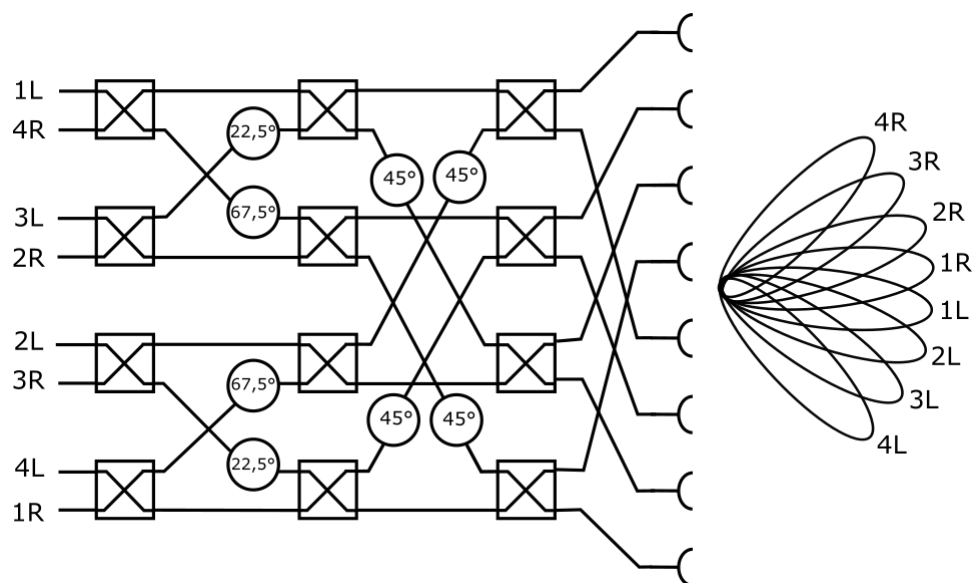
Z tvaru charakteristik již lze soudit vhodnost použití jednotlivých funkcí okna. Binomické pole sice odstraňuje postranní laloky, ale příliš rozšiřuje hlavní lalok a anténa tak ztrácí směrovost. Čebyševovo okno má téměř shodnou charakteristiku s Taylorovým. V případě Taylorova je však mírně vyšší zisk a více potlačené postranní laloky.

Zobrazené charakteristiky byly simulovány na anténě s 8 prvky, která není pro názornost nejvhodnější. Pro dosažení lepších výsledků by bylo vhodné využití větší antény. Z porovnání vychází jako nejlepší Taylorovo okno. Při testování tvarovače bude však využito Čebyševovo okno. V prostředí GNU Octave, které je při návrhu použito není totiž Taylorova funkce dostupná.

1.2 Způsoby tvarování svazků

1.2.1 Butlerova matice

Nejstarším způsobem řízení tvarování svazku je použití pevně zapojené matice hybridních spojek a fázovacích členů [18]. Na jedné straně matice jsou vstupy pro jednotlivé svazky, na druhé jsou prvky antény. Na obrázku 1.9 je znázorněno zapojení matice včetně vytvořených svazků.



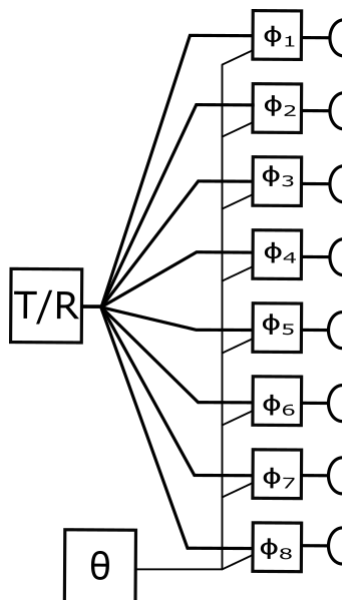
Obr. 1.9: Zapojení Butlerovy matice

Aplikace signálu na jeden ze vstupů způsobí vybuzení prvků antény s fázovými posuny tak, že je vytvořen svazek příslušný k použitému vstupu. Totéž platí u příjmového signálu, kde jsou na příslušných výstupech dostupné příjmové svazky. Výhodou je možnost buzení více svazků současně.

Využití je nejčastější v aplikacích, kde není požadavek na změnu polohy svazku jako například u mobilních vysílačů. V radarové technice bývá Butlerova matice někdy kombinována s elektronickým vychylováním svazku.

1.2.2 PESA

Pojem passive electronically scanned array (PESA) označuje pasivní fázovanou řadu [17]. Jedná se o anténní řadu, kterou je možné elektronicky vychylovat. Všechny elementy antény jsou přitom napojeny na jeden vysílač a jeden přijímač. Samotné vychylování probíhá pomocí elektronicky ovladatelných fázovacích článků předřazených prvkům antény. Konfigurace je vyobrazena na obrázku 1.10.

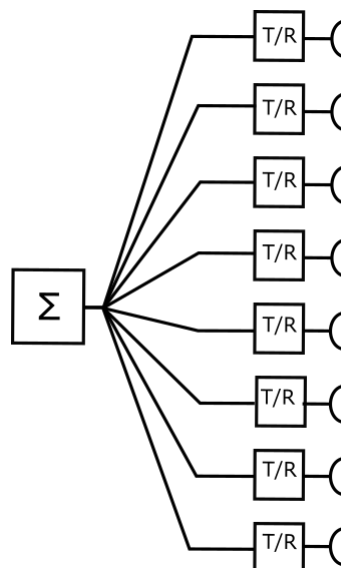


Obr. 1.10: Blokové schéma PESA

V porovnání s Buttlerovou maticí je PESA univerzálnější, ale může pracovat pouze s jedním svazkem.

1.2.3 AESA

Aktivní řada (active electronically scanned array) koncepčně navazuje na PESA a její vznik byl umožněn díky rozvoji integrované elektroniky [17]. Každý prvek antény je vybaven integrovaným vysílacím a přijímacím modulem, který je možné elektronicky ovládat (obrázek 1.11).

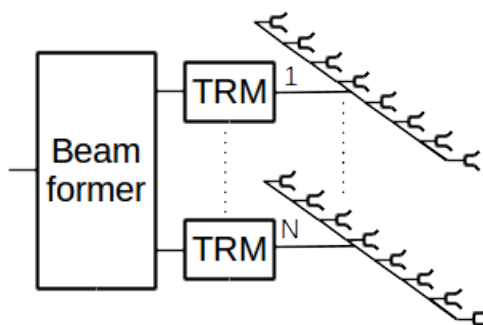


Obr. 1.11: Blokové schéma AESA

Hlavními výhodami jsou schopnost práce s více svazky i frekvencemi, odolnost vůči poruše části prvků a menší zkreslení signálu při distribuci do antény. Dnešní technologie navíc umožňují digitalizaci signálu přímo v T/R modulech a celý radar je tak možné řídit číslicově.

AESA v jedné rovině

Anténní řada je složena z pasivních subřad kontrolovaných jedním vysílacím a přijímacím členem (obrázek 1.12).



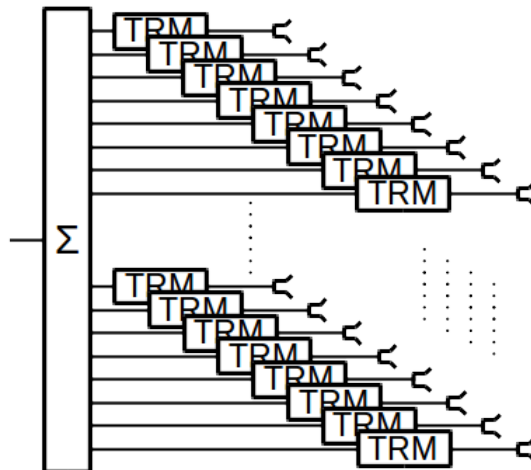
Obr. 1.12: Anténa s elektronicky řízeným svazkem v jedné ose
(použito s laskavým svolením P. Šedivého)

Vychylování svazku je možné pouze v jedné ose zatímco ve druhé je vyzařovací charakteristika pevná. Výhodou je jednoduchá konstrukce antény a s tím související cena. Tato konfigurace bývá používána pro radarové systémy s rotující anténou,

kde je vychylování použito pro měření elevace, zatímco azimut je zjištěn z natočení antény.

Plná AESA

Anténní řada je tvořena maticí jednotlivých elementů. Každý prvek je přitom vybaven vlastním T/R modulem (obrázek 1.13).

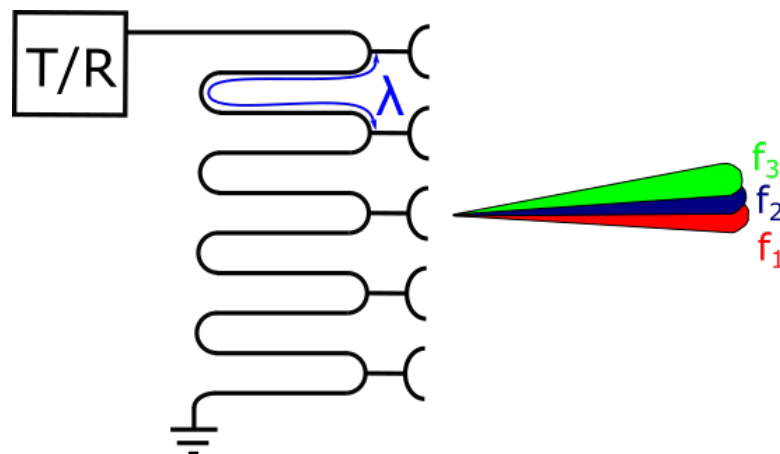


Obr. 1.13: Konfigurace antény s řízením v obou osách
(použito s laskavým svolením P. Šedivého)

Hlavní výhodou je možnost vychylování v obou osách. Využívá se u rotujících i statických antén radarů.

Frekvenční disperze

Speciálním případem fázované anténní řady je frekvenčně závislá anténa. Ta je tvořena řadou prvků spojených v sérii.



Obr. 1.14: Frekvenčně závislá anténa

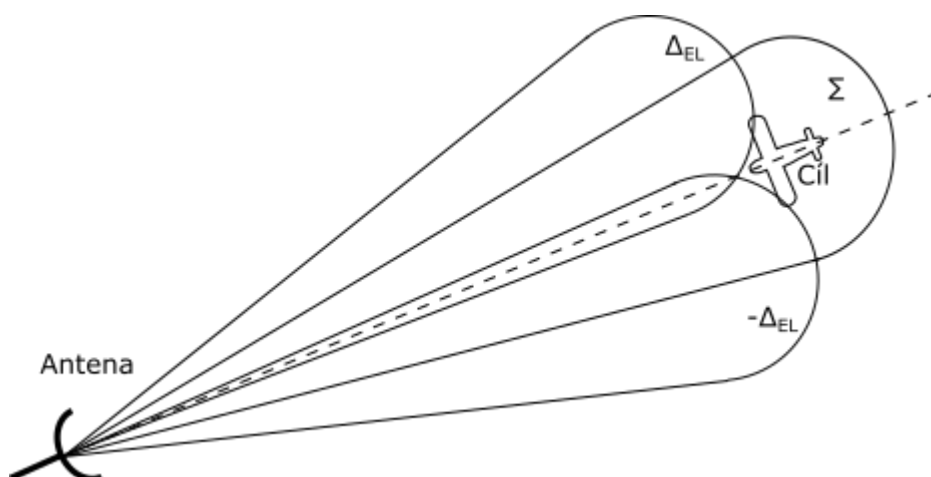
Délka dráhy mezi jednotlivými elementy je konstantní a je navržena tak, aby při pracovní frekvenci byl fázový posun mezi prvky vždy 360° . Změnou frekvence vysílání dojde ke změně fázového posuvu mezi prvky a tím dochází k vychýlení svazku (obrázek 1.14) [5].

I přes zdánlivou jednoduchost má tato konstrukce řadu omezení, kvůli kterým není u moderních radarů využívána.

1.3 Monopulzní určení polohy cíle

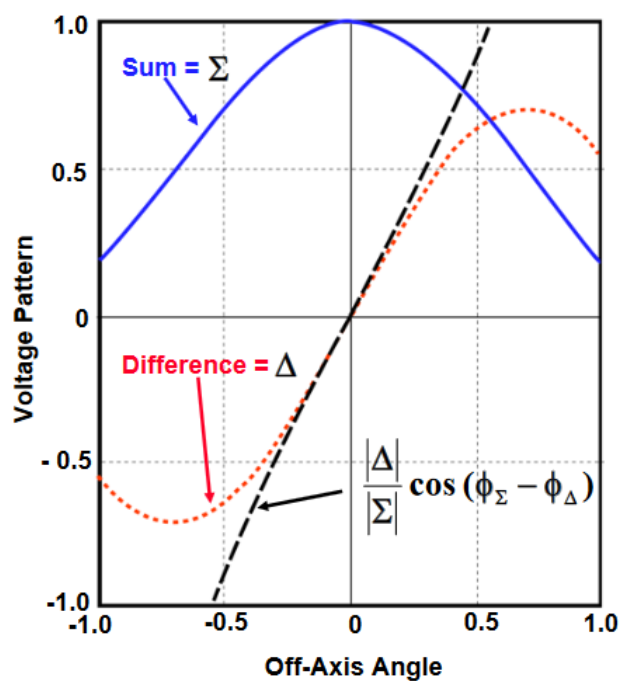
Radar při prohledávání prostoru zaznamená cíl ve chvíli, kdy vstoupí do jeho hlavního laloku. Při určování úhlové polohy je však předpokládáno, že je cíl v ose svazku antény. Tím vzniká chyba o velikosti až poloviny šířky svazku. Dříve se přesnější měření úhlu řešilo pohybem svazku v okolí cíle a průměrováním maximálního a minimálního úhlu, pro který byl cíl zaznamenán. Tato metoda však byla zatížena velkou chybovostí vlivem šumu a pohybu cíle mezi pozorováními [5].

U novějších radarů se pro upřesnění úhlové odchylky využívá monopulzní určování polohy, kde je poloha cíle je určena z jednoho pulzu a vyhodnocení je tak rychlejší a odolné vůči rušení. Princip monopulzu spočívá v rozdělení anténní řady na 2 poloviny. Při vysílání pracují obě poloviny současně zatímco na příjmu jsou brány jako samostatné antény. Z přijatého signálu je vytvořen součtový signál Σ a rozdílový signál Δ_{AZ} vzniklý vzájemným odečtením polovin antény. Rozdílový signál je tvořen částečně se překrývajícími svazky. Směr je obvykle volen tak, aby se překrývaly v místě s poloviční intenzitou -3dB (obrázek 1.15) [12].



Obr. 1.15: Monopulzní detekce cíle

Bude-li cíl přímo v ose svazku, pak bude signál obou částí rozdílového signálu stejný. Odchylka cíle od osy způsobí nepoměr mezi polovinami rozdílového signálu, ze kterého je možné vypočítat úhel odchylky. Průběhy součtového a rozdílového signálu včetně výpočtu odchylky jsou znázorněny na obrázku 1.16. Čím strmější bude poměr rozdílového a součtového signálu, tím přesnější bude odhad chyby úhlu.



Obr. 1.16: Vyhodnocení úhlové odchylky [13]

Zlepšení přesnosti amplitudové monopolní syntézy je možné zajistit zvýšením strmosti rozdílového signálu. To se provádí aplikací váhové funkce jednotlivé prvky antény. Váhová funkce je volena tak, aby prvky vzdálenější od středu, které mají na přesnost větší vliv, měly větší váhu [12].

Monopolní detekci je rovněž možné využít současně s vychylováním svazku pro dosažení vyšší přesnosti.

1.4 Vliv reálné signálové cesty

V případě skutečného radaru je nutné při vychylování a tvarování svazku započítat i změnu amplitudy a fáze vzniklou vlivem nedokonalosti reálných signálových cest. Tuto změnu je nutné v rámci systému měřit a kompenzovat, aby se neprojevila na výsledcích měření. Vzhledem k tomu, že většina signálového zpracování probíhá v číslicové podobě, řeší se pouze cesta mezi anténou a digitalizačním modulem.

Signálovou cestu si lze zjednodušeně představit jako vícebran. Z teorie elektroniky je známé, že chování mnohabranu lze popsat skupinou parametrů, získanou reálným měřením daného systému. K tomuto účelu většinou slouží Impedanční Z-parametry, Admitanční Y-parametry nebo Hybridní H-parametry. Tyto parametry pracují s proudy a napětími a definují chování systému v případech kdy je jedna z bran naprázdno nebo nakrátko. U systémů pracujících na vyšších, typicky mikrovlnných frekvencích toto není možné, neboť zpravidla není možné dosáhnout stavu naprázdno nebo nakrátko, aniž by docházelo k odrazům a interferenci vln. Dalším problémem je nestálý poměr napětí a proudu v různých místech systému. Z toho důvodu se pro popis vysokofrekvenčních zařízení využívají s-parametry [19], které namísto proudů a napětí řeší průchod normalizované zdrojové vlny systémem a její odrazy. Pro každou cestu mezi dvojicí bran, tvořených zpravidla koaxiálním vedením nebo vlnovody, existuje jeden parametr popisující přenos mezi těmito branami. Systém s N branami tak obsahuje N^2 parametrů. s-parametry jsou komplexní čísla často v exponenciálním tvaru, představující amplitudu a fázi signálu.

Pro stanovení s-parametrů jsou definovány 2 standardizované vlny definované napětím, proudem a impedancí brány. Vlna působící na bránu i (rovnice 1.10)

$$a_i = \frac{V_i + Z_i I_i}{2\sqrt{|Re(Z_i)|}} \quad (1.10)$$

a vlna odražená od i -té brány (rovnice 1.11)

$$a_i = \frac{V_i - Z_i^* I_i}{2\sqrt{|Re(Z_i)|}}, \quad (1.11)$$

kde Z_i^* je komplexně sdružené číslo k impedanci. Pro stanovení s-parametrů pak platí:

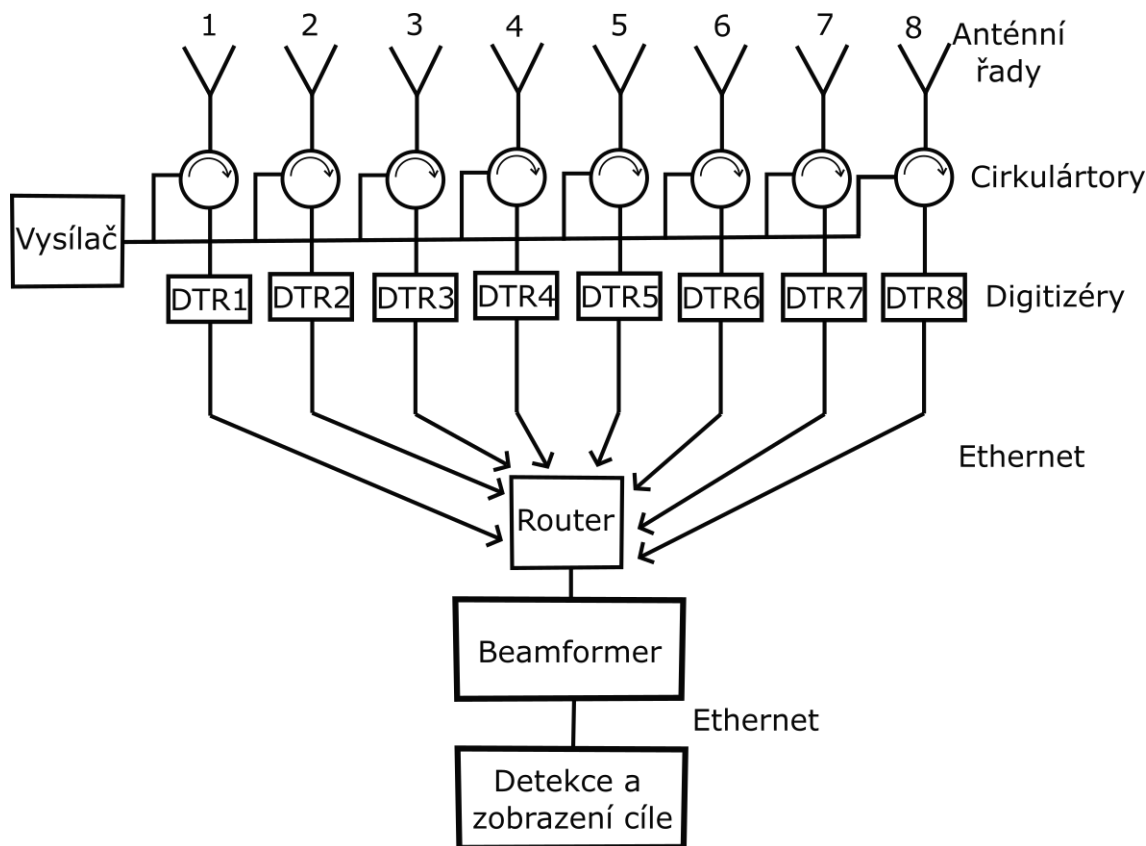
$$\begin{aligned} s_{11} &= \left. \frac{b_1}{a_1} \right|_{a_2=0} \\ s_{12} &= \left. \frac{b_1}{a_2} \right|_{a_1=0} \\ s_{21} &= \left. \frac{b_2}{a_1} \right|_{a_2=0} \\ s_{22} &= \left. \frac{b_2}{a_2} \right|_{a_1=0} \end{aligned} \quad (1.12)$$

Pro měření s-parametrů se využívá síťový analyzátor. Přístroj vybudí testovaný systém vysláním zdrojové vlny a na výstupu zaznamená průchozí vlnu. Síťové analyzátoři mohou měřit buď skalárně, kdy zaznamenají pouze amplitudy vln, nebo vektorově se záznamem komplexních parametrů [19].

Do tvarovače svazků s-parametry vstupují jako další sada koeficientů, kterými je třeba při výpočtu svazků dělit vstupní data. Pro každý kanál svazku je přitom nutné zvolit jinou sadu s-parametrů podle cesty zdrojového signálu.

2 Integrace tvarovače do radaru

Navrhovaný tvarovač je určen pro radar AESA s vychylováním v elevaci. Blokové schéma radaru s důrazem na znázornění začlenění tvarovače je na obrázku 2.1.



Obr. 2.1: Blokové schéma demonstrátoru

Na začátku periody vysílací modul (TX) vygeneruje pulz, který odešle do anténní řady. Během jedné periody jsou vysílány 2 pulzy: SP(krátký pulz) pro blízké cíle a LP(dlouhý pulz) pro cíle vzdálené. Odražený signál je veden z antény do přijímacích modulů DTR (pro každou řadu jeden). Spojení mezi anténou, vysílacím a přijímacím modulem je tvořeno cirkulátorem. Ten si lze představit jako mechanický multiplexor, tvořený vlnovodem ve tvaru Y s feritovým terčem uprostřed. Vlivem působení vnějšího magnetického pole vzniká v okolí feritu gyotropní prostředí a signál je veden pouze jedním směrem (např. po směru hod. ručiček), což zajistí, aby se vysílaný signál dostal pouze na anténu a přijatý naopak z antény do přijímače [13].

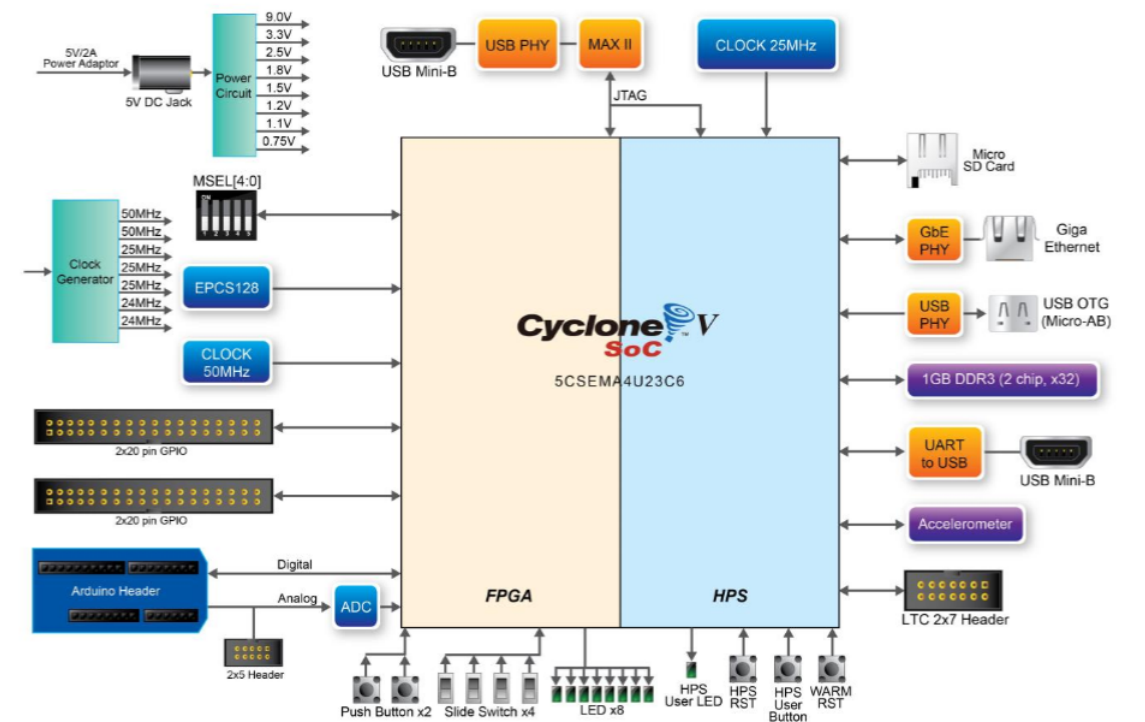
DTR modul zajišťuje navzorkování a převod signálu A/D převodníkem. Další komunikace je v rámci celého radaru řešena po sběrnici Gigabit Ethernet. DTR moduly odesílají data řad ve standardizovaném paketu pomocí UDP protokolu.

Při zpracovávání je třeba dbát na bytovou organizaci. Data hlavičky mají organizaci *big endian* zatímco data samotná *little endian*.

Data řad jsou přijata tvarovačem (beamformer) a přepočítána na jednotlivé svazky, které jsou opět přes UDP protokol odeslána obvodům pro detekci cíle. Svazků může být libovolné množství, dané konfigurací detektoru cíle. Požadavek je, aby byl počet a směr výsledných svazků co nejlépe konfigurovatelný. Po detekování cílů jsou výsledná data vykreslena na monitor.

2.1 Vývojová deska

Vzhledem k nedostupnosti vhodnějšího hardwaru bude návrh beamformeru testován na vývojovém kitu DE0-nano od firmy Terasic. K otestování algoritmu tvarování by tento kit měl být dostačující, ale pro praktické použití je nezbytné jeho nahrazení výkonnějším hardwarem. Kit DE0-nano/Atlas Soc je univerzální vývojová platforma pro práci s obvody System on chip firmy altera. Deska je postavena okolo obvodu rodiny Cyclone V s důrazem na nízkou cenu a je určena především pro seznámení s obvody SoC FPGA. Blokové schéma desky s popisem hlavních částí je na obrázku 2.2.

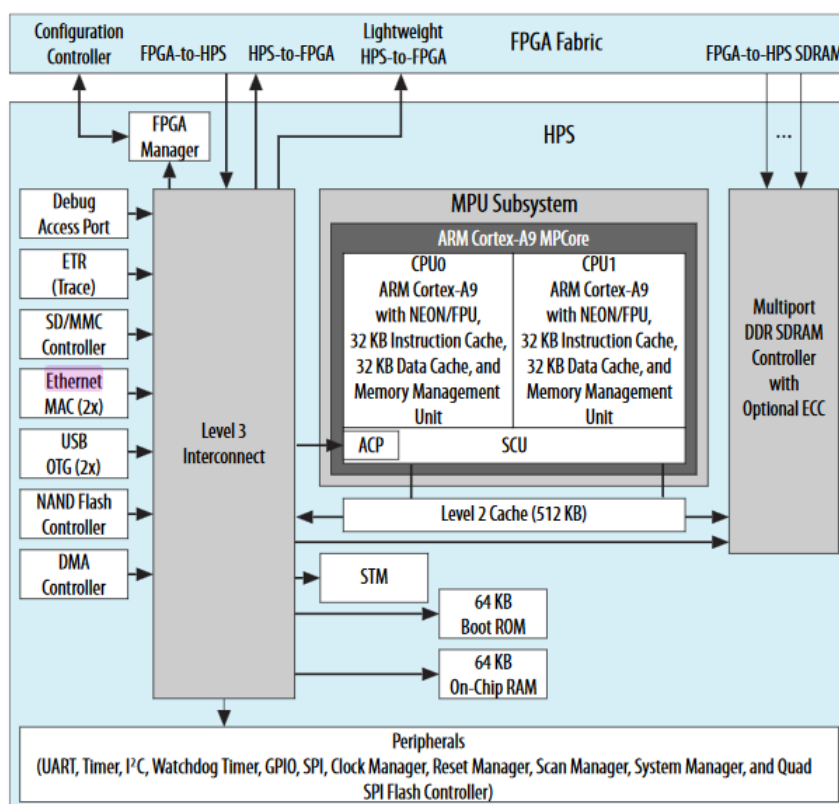


Obr. 2.2: Vývojový kit DE0-nano [20]

Kit kromě řídicího obvodu disponuje řadou periférií potřebnou pro běžné aplikace. Mimo standardních částí jako jsou napájecí obvody, generátor hodinového cyklu, tlačítek, LED diod, I/O pinů můžeme na desce najít například i A/D převodník, RAM paměť, USB Host, Akcelerometr, nebo Fyzickou vrstvu pro Ethernet. Program je do FPGA obvodu možné nahrát několika způsoby. Způsob je volen sadou přepínačů MSEL a je možné vybrat nahrávání přes JTAG, využití konfiguračního obvodu EPCS128 nebo načtení ARM procesorem z SD karty.

2.1.1 FPGA Cyclone V

Cyclone V je cenově nejdostupnější rodina obvodů firmy Intel. Mezi nabízenými obvody jsou nejen klasická FPGA, ale i SoC kombinující FPGA s hard IP ARM procesory. Jedním z nich je i 5CSEMA4U23C6N, který je obsažený na vývojovém kitu. Ten kromě FPGA obsahuje rovněž hardwarový dvoujádrový procesor s jádrem ARM Cortex A9. Blokové schéma procesoru je na obrázku 2.3.



Obr. 2.3: Blokové schéma HPS [21]

Pro funkci tvarovače je důležité využití sběrnice Ethernet. Fyzická vrstva rozhraní osazená na desce kitu je přes sběrnici RGMII vedena do HPS části SoC

obvodu, kde je napojena na řadič Ethernet MAC s přímým přístupem do paměti procesoru. Pro usnadnění práce běží na HPS Linux distribuce Angstrom obsažená na paměťové kartě kitu, která se stará o obsluhu periférií. Díky tomu je možné při tvorbě programu využívat vyšších funkcí jazyka C/C++.

Kvůli využití slabšího hardware nebude pro zpracování UDP paketů možné dosažení požadované rychlosti 1Gbit/s a proto bude rychlost pro účely testování snížena na 10Mb/s. Nahrazení přijímací části HPS by nemělo mít vliv na realizaci výpočetní části, která je hlavním cílem této práce. Ve finální aplikaci pak bude funkce HPS nahrazena pomocí IP jader 1Gbit nebo 10Gbit Ethernetu a pomocnými obvody, které by měly umožnit plnou rychlost.

Procesorová část SoC obvodu je propojena s FPGA architekturou pomocí HPS-FPGA můstků tvořených AXI sběrnicí. Ty v plné verzi podporují 32,64 a 128bitovou komunikaci. K dispozici je rovněž odlehčená verze s menším zpožděním, ale pouze 32bitovou šířkou.

FPGA část obvodu je napájena zvlášť a může být pro účely úspory energie nezávisle vypnuta. Základním stavebním prvkem architektury je Adaptivní logický modul (ALM), který FPGA Intel využívají namísto jednoduchých LUT. ALM obsahuje kromě 8-vstupé LUT ještě dvě 2 bitové úplné sčítačky a 4 registry. V tabulce je vypsán přehled dostupných zdrojů obvodu [21].

Tab. 2.1: Dostupné zdroje obvodu 5CSEMA4U23C6N [21]

ZDROJ	Dostupný počet
Logické elementy	40 000
ALM	15 880
Registry	60 376
DSP Blok s proměnnou přesností	84
Násobička 18x18	168
GPIO	145

Pro implementaci tvarovače je klíčový dostatečný počet hardwarových 16-bitových násobiček. Těch obvod nabízí 168 samostatně a dále pak 2 v každém DSP bloku, kterých je v obvodu obsaženo 84. Celkem je tedy možné využít 336 16-bitových násobiček.

3 Návrh tvarovače

3.1 Realizovaná funkce

Cílem navrženého algoritmu je realizace výpočtu 3 rovnic, které jsou odvozeny na základě předchozích kapitol pro součtový azimutální a elevační kanál [22]:

$$B_{\Sigma}(k) = \sum_1^N \frac{a_n(k)}{s_{(n)DRU\Sigma S-A\Sigma}} \cdot S_{\Sigma}(n), \quad (3.1)$$

$$B_{\Delta A}(k) = \sum_1^N \frac{a_n(k)}{s_{(n)DRU\Delta S-A\Delta}} \cdot S_{\Delta}(n), \quad (3.2)$$

$$B_{\Delta E}(k) = \sum_1^N \frac{b_n(k)}{s_{(n)DRU\Sigma S-A\Sigma}} \cdot S_{\Sigma}(n), \quad (3.3)$$

kde: B_{Σ} je signál součtového svazku,

$B_{\Delta A}$ je signál rozdílového azimutálního svazku,

$B_{\Delta E}$ je signál rozdílového elevačního svazku,

a_n je komplexní tvarovací váha n -té řady v azimutu,

b_n je komplexní tvarovací váha n -té řady v elevaci,

s_n jsou s -parametry n -té řady od fázového středu součtového/rozdílového kanálu k převodníku v DRU,

$S_{\Sigma}(n)$ je signál součtového kanálu n -té řady,

$S_{\Delta}(n)$ je signál rozdílového kanálu n -té řady,

Na rozdíl od odvození je v praktické aplikaci tvarovací váha navíc dělena s-parametry (viz kapitola 1.4) signálové cesty mezi výstupy antény a DTR modulem. Komplexní váhy jsou vypočteny z rovnice 3.4 vycházející ze vztahu 1.3.

$$a_n = X(n) \cdot e^{j2\pi \cdot (n:N-1) \cdot \frac{\Delta L_v}{\lambda} \cdot \sin(\theta_k - \theta_0)}, \quad (3.4)$$

kde $X(n)$ je vektor funkce okna

n je číslo anténní řady $1 \dots N$,

ΔL_v je rozteč prvků řady,

λ je vlnová délka signálu,

θ_k je úhel požadovaného svazku,

θ_0 je odklon apertury

V rovnici pro elevaci se uplatňuje rozšířená komplexní váha pro zvýšení přesnosti monopolzního zpracování, která zvyšuje význam krajních elementů řady (rovnice 3.5).

$$b_n = 2 \cdot a_n \cdot \left(\frac{2n - N - 1}{N - 1} \right) \quad (3.5)$$

kde a_n jsou tvarovací váhy z rovnice 3.4,

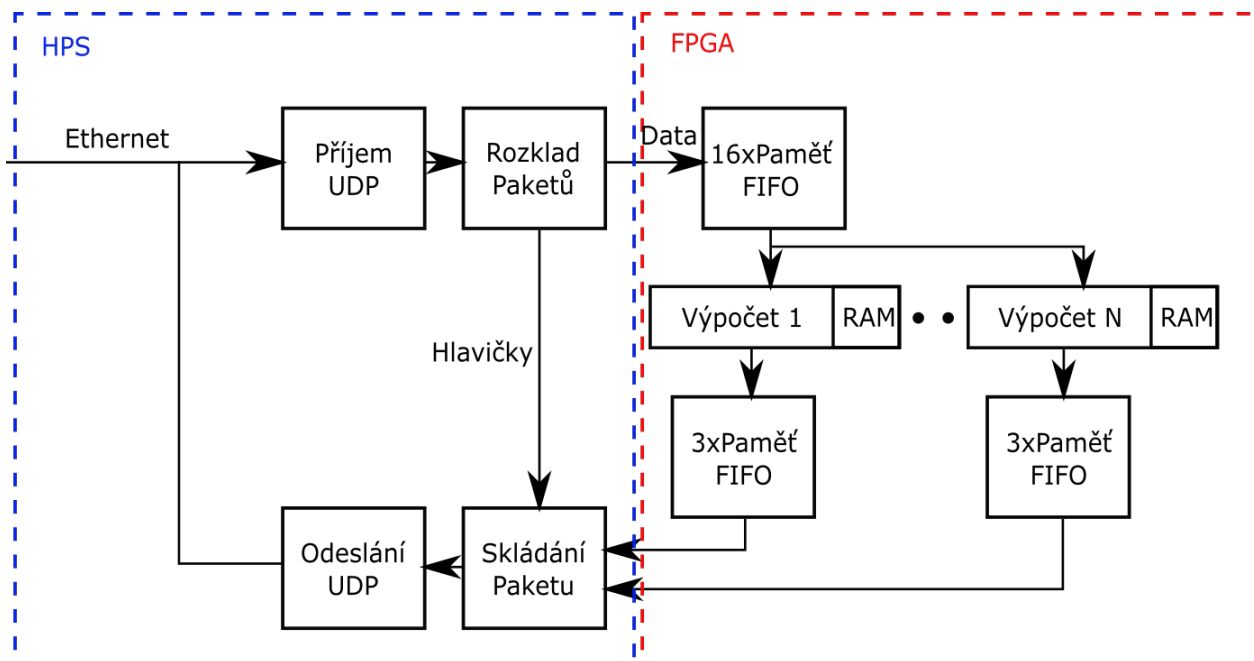
n je číslo prvku anténní řady odpovídající indexům,

N je celkový počet prvků řady

3.2 Způsob implementace

Princip funkce spočívá v příjmu dat z DTR modulů jednotlivých řad, výpočet požadovaného počtu svazků a odeslání dat k vyhodnocení polohy cíle.

Části implementace byly v obvodu SoC rozděleny mezi HPS a FPGA. Procesorová část, která je vybavena hardwarem pro práci se sběrnici Ethernet zajišťuje příjem, zpracování a odesílání datových paketů, FPGA zajišťuje paměťový prostor a paralelní výpočty. Provedení implementace je patrné z blokového schématu na obrázku 3.1.



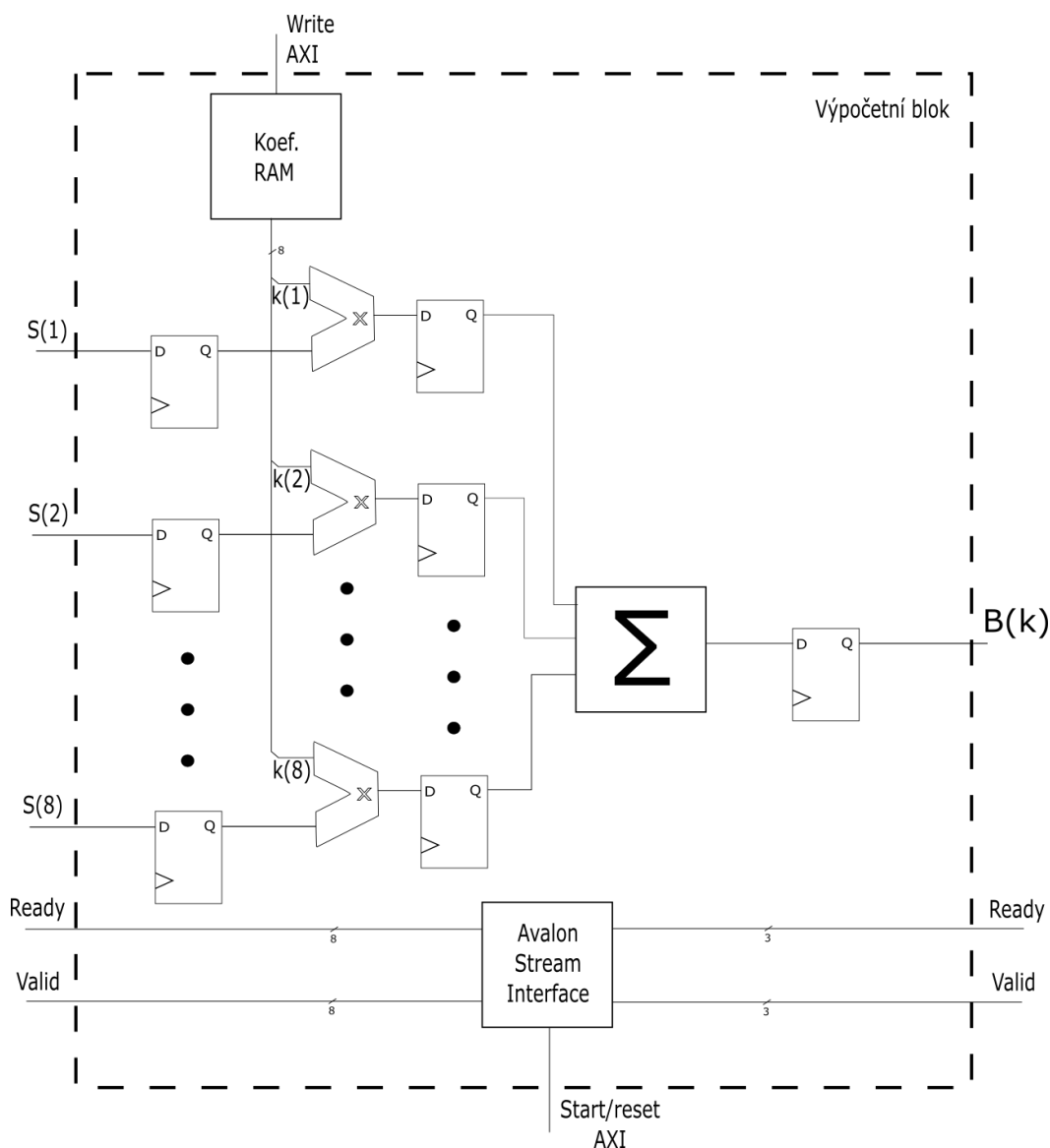
Obr. 3.1: Blokové schéma tverovače

Na vstupu jsou přijaté UDP pakety. Ty jsou při zpracování dále rozděleny na hlavičku a data. Hlavičky slouží k roztřídění dat a k identifikaci výstupních svazků. Data jsou ukládána do příslušných pamětí FIFO. Pro každou z anténních řad jsou určeny 2 paměti uchovávající data součtového a rozdílového kanálu. Po přijetí kompletní sady dat tvarovač vypočte svazky s parametry danými koeficienty v paměti RAM. Výsledné svazky jsou uloženy do výstupních pamětí FIFO. Ty jsou vždy ve skupinách po třech pro součtový, azimutální a elevační kanál. Z výstupních pamětí jsou data po vyčtení složena do UDP paketů s přidáním původních hlaviček a po doplnění údajů o parametrech svazku je celý paket odeslán. Ze schématu na obrázku 3.1 je také patrné rozmístění návrhu v HPS a FPGA částech obvodu. Jednotlivé bloky budou podrobně popsány v následujících podkapitolách.

3.2.1 Výpočetní blok

Výpočet se skládá ze 3 hlavních částí: sítě sčítaček a násobiček, paměti RAM pro uložení koeficientů a bloku doplňujícího signály rozhraní Avalon stream. Výpočet rovnice jednoho kanálu je realizován zapojením 8 násobiček a 8-vstupé sčítačky. Blokové schéma je na obrázku 3.2. Schéma bylo záměrně zjednodušeno. Po implementaci pracují všechny části výpočtu s komplexními čísly.

Vstupy $S(n)$ jsou data přijatá jednotlivými řadami (z rovnic 3.23.33.1). $K(n)$ jsou



Obr. 3.2: Blokové schéma výpočtu 1 kanálu

koeficienty uložené v paměti RAM. Do RAM je možné zapisovat přes integrované rozhraní AXI sběrnice.

Po vynásobení vstupů koeficienty a provedení součtu mezivýsledků je výstupem 1 kanál $B(k)$ počítaného svazku. Celý svazek je složen ze 3 kanálů (součtový, azimutální, elevační).

Blok Streamového rozhraní doplňuje výpočet o signály *ready* a *valid* nezbytné ke komunikaci s okolím. Blok je možné ovládat pomocí AXI sběrnice pro řízení celého výpočtu.

Modifikovatelnost návrhu

Jedním z požadavků na návrh je jeho modifikovatelnost. Ta je zajištěna parametrickým návrhem komponent. Pro snadnou změnu parametrů návrhu jako bitovou šířku vstupních dat nebo počet řad antény jsou veškeré parametry přesunuty do balíku *beamformer.pkg*. Kromě globální definice parametrů použití balíku umožňuje definici vektorových polí, které lze dále použít jako porty jednotlivých komponent. Modifikovatelnost návrhu je bohužel omezena použitím programu *Qsys* pro sestavení vrcholné hierarchie. Ačkoli je tento program preferován pro popis top-level architektury, pro obvody Intel jeho kompilátor pracuje pouze s minimem funkcí VHDL a neumožňuje využití parametrů. Návrh je proto zabalen do souboru *qsys_top*, který veškeré generické části nahrazuje pevnou šířkou. Změna parametrů tvarovače je tak ještě podmíněna změnou šířek v tomto souboru.

Realizace násobení

Koeficienty jsou tvořeny komplexními váhami spojenými s s-parametry. Samotné násobení probíhá v algebraickém tvaru podle rovnice 3.6.

$$(a_1 + b_1 i) \cdot (a_2 + b_2 i) = (a_1 \cdot a_2 - b_1 \cdot b_2) + (a_1 \cdot b_2 + b_1 \cdot a_2) i \quad (3.6)$$

Výpočet obsahuje 4 operace násobení. Počet potřebných HW násobiček tedy bude $4 \cdot 8 = 32$ pro jeden kanál. Pro výpočet jednoho paprsku je nutné počítat 3 kanály (součtový, azimutální, elevační). Celý paprsek tedy vyžaduje $3 \cdot 32 = 96$ HW násobiček. Při využití daného obvodu je tak možné počítat maximálně tři paprsky paralelně.

Realizace součtu

Při součtu komplexních čísel v algebraickém tvaru se sčítají reálné a imaginární části zvlášť. Díky tomu je možné využít dvou jednoduchých 32-bitových sčítaček.

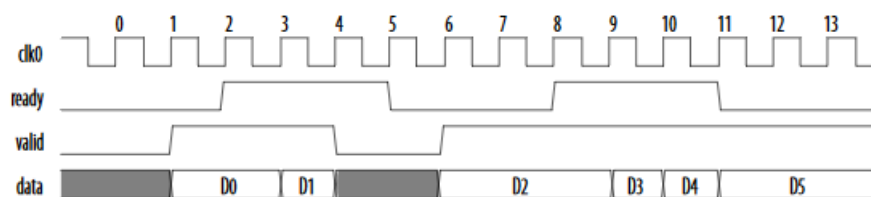
Sčítání 8 32-bitových čísel s 32-bitovým výsledkem, které je ve tvarovači použito, není úplně správné vzhledem k poměrně vysoké pravděpodobnosti přetečení výsledku. Jedním z možných řešení prevence přetečení by bylo zvětšení bitové šířky výsledku. Pro odstranění rizika přetečení i při extrémních hodnotách všech vstupů by bylo nutné rozšířit výsledek o 3 bity. Pro finální řešení v obvodu Aria 10, kde bude za sčítačkou zařazen blok pro převod na *float* s jednoduchou přesností je toto rozšíření plánováno. U testovacího obvodu cyclone V zůstává přetečení nešetřeno. Hlavním důvodem je šířka sběrnice (můstku), zajišťující

přenos dat mezi HPS a FPGA, která je 32 bitů. Výsledek o šířce 35 bitů by tak nebylo možné efektivně vyčítat z paměti. Rozšíření na 64 bitů by umožnilo vyčítání ve dvou krocích, ale vyžadovalo by zvětšení výstupních pamětí FIFO a tím i náročnost na zdroje obvodu. Chod testovací verze tvarovače bez ošetření přetečení je možný vzhledem k charakteru vstupních dat. Ta ačkoli přichází ve formátu *Int16*, nepřesahuje hodnotami rozsah 8 bitů. Po násobení 16-bitovými koeficienty tedy výsledky mají reálně 24 bitů a je možné je sčítat do 32 bitového výsledku bez přetečení. Pro reálnou aplikaci, kde nebude zaručena rezerva ve vstupních datech, je ošetření nutné.

Avalon Stream

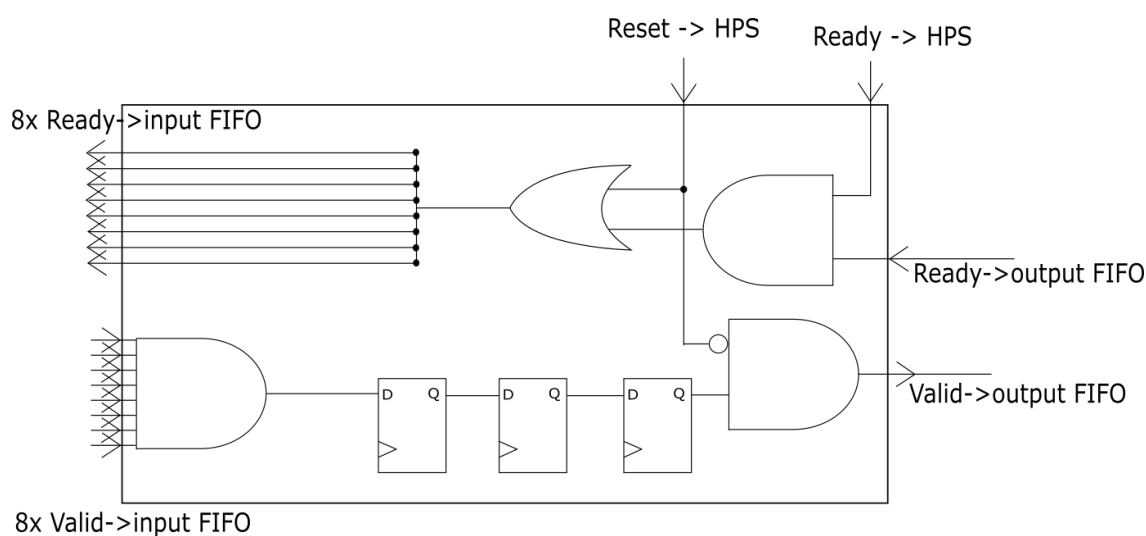
Vstupní data pro výpočet musí být dodávána pokud možno co nejrychleji formou streamu, aby s každým hodinovým taktém byla zpracována jedna hodnota. Stejně by tomu mělo být i v případě výstupu, který bude pouze o několik taktů opožděn vlivem registrů vložených do výpočtu pro synchronní chod. Zároveň je nutné zajistit snadné propojení nejen s HPS použitým pro demonstraci, ale i s jeho náhradou ve formě IP jader. Rozhraní výpočetního modulu je nutné rozšířit o signály, které zajišťují kontrolu nad přenosem streamu dat. Jako vhodné se jeví využití standardizovaného rozhraní Avalon Streaming®, jímž disponuje většina jader vyvinutých firmou Intel [25].

Rozhraní Avalon streaming je vhodné pro jednosměrnou komunikaci s malým zpožděním. Zároveň je možné v případě potřeby využít i složitějších funkcí pro řízení provozu na sběrnici. Komunikace má vždy 2 účastníky - vysílač (source) a přijímač (sink). Pokud má vysílač aktivní vstup *ready*, s hodinovým signálem odesílá data na sběrnici a signálem *valid* potvrzuje jejich platnost. Přijímač data zachytává a v případě neschopnosti dalšího příjmu zabrání shozením signálu *ready* dalšímu vysílání. Časový průběh je na obrázku 3.3.



Obr. 3.3: Časový průběh komunikace na sběrnici Avalon Stream [25]

Pro návrh to znamená rozšíření vstupů a výstupů výpočetního modulu o 8 výstupních signálů *ready* a 1 signál *valid*. Výstupní signál *valid* je aktivní v případě, kdy jsou aktivní všechny vstupní *valid* signály, přicházející ze vstupních pamětí FIFO. Pro správnou funkci je signál *valid* zpožděn o takový počet registrů v signálové cestě, aby jeho stav odpovídal platným vstupům. Signály *ready* jsou odvozeny z *ready* signálů výstupních pamětí FIFO. Navíc jsou přidány signály *start* a *reset* napojené na výstupy HPS pro spouštění výpočtu a mazání pamětí FIFO. Realizace rozhraní pro avalon stream je znázorněna na obrázku 3.4.



Obr. 3.4: Realizace ovládacích signálů sběrnice Avalon stream

Paměť RAM

Paměť pro koeficienty je tvořena polem registrů pro uchování jednotlivých hodnot. Ačkoli je paměť popsána v jazyce VHDL, pro její tvorbu byl vzhledem k jednoduchosti návrhu, ale zdlouhavému zápisu použit automatický generátor na bázi jazyka Python. Ten je po nastavení parametrů registrového pole schopen generovat nejen registrové pole, ale i jeho napojení na AXI sběrnici. Dále je schopen generovat i dokumentaci ve formátu HTML nebo hlavičkové soubory pro práci s pamětí napojenou na IP core procesory.

Samotný generátor byl dodán firmou Retia a.s. Pravděpodobně se však jedná pouze o modifikaci open-source projektu *hdlregs* dostupného například na platformě Github.

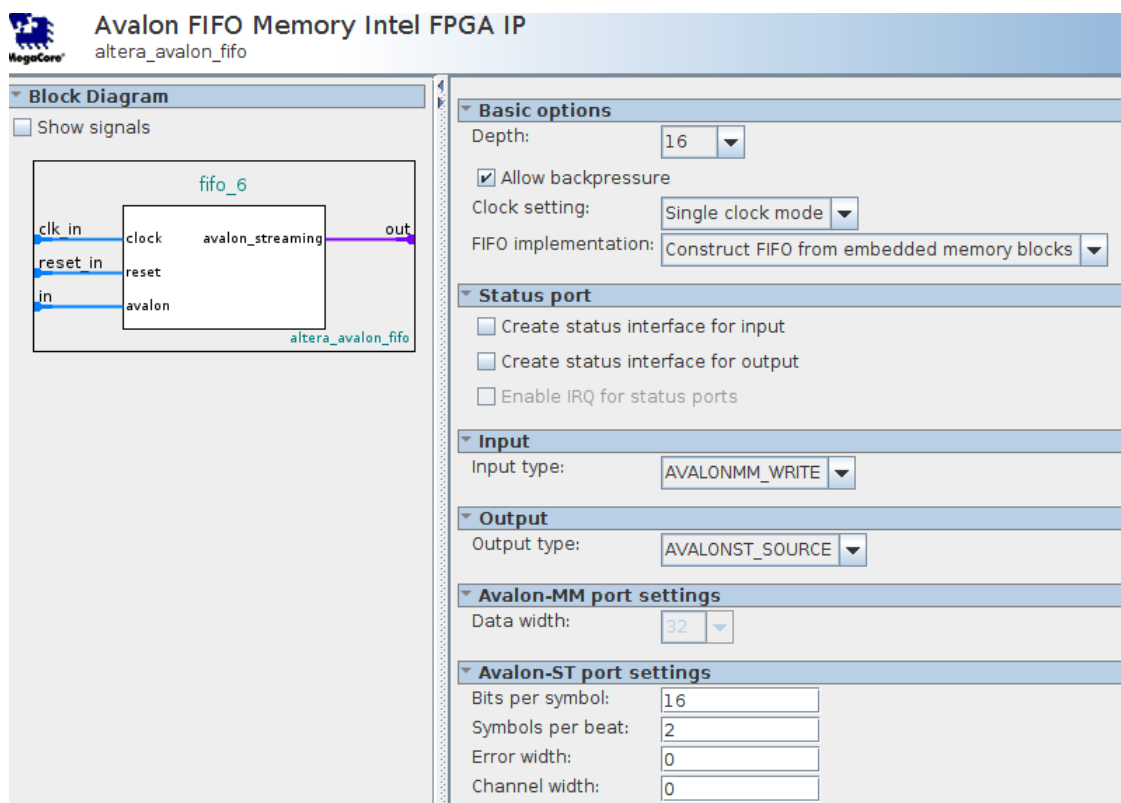
3.2.2 Paměť FIFO

Jako rozhraní mezi HPS a výpočtením modulem je nutné využít paměť, do které je možné uložit data až do doby přijetí paketů z DTR modulů všech řad. Datové pakety obsahují vždy celou sadu dat, která se zpracovává postupně a proto je vhodné využití paměti FIFO. Tuto paměť je možné vytvořit ručně pomocí VHDL kódu.

Jednodušším řešením je však využití již hotových IP jader, které nabízí firma Intel v rámci vývojového prostředí Quartus. Tato jádra mají kromě ulehčení práce ještě tu výhodu, že disponují standardizovanými rozhraními Intel Avalon®, která umožňují propojení mezi různými IP jádry [23]. Využití takovéto paměti umožní pozdější nahrazení HPS IP jádrem bez nutnosti modifikace výpočetní části. Další drobnou výhodou je také fakt, že Intel ke svým jádrům dodává rovněž knihovny pro jejich využití v rámci HPS.

Pro vytvoření paměti je ve vývojovém prostředí dostupný jednoduchý konfigurator (obrázek 3.5). Hlavními parametry jsou bitová šířka slova, počet slov na takt, délka paměti a počet hodinových vstupů. Důležité je také povolení zpětného tlaku, které zabrání ztrátě dat zastavením zápisu v případě přeplněné paměti. Volitelně je možné přidat rozhraní pro zjišťování stavu paměti.

Pro tvarovač je zvolena paměť s šířkou 32 bitů, jedním zdrojem hodinového signálu a stavovým rozhraním na vstupu pro kontrolu zaplnění paměti. Vstupní rozhraní pro spojení s HPS je *Avalon memory map*, které zajišťuje přímé napojení do adresního prostoru procesoru. Na výstupu je pak rozhraní Avalon Streaming pro spojení s výpočtním modulem.



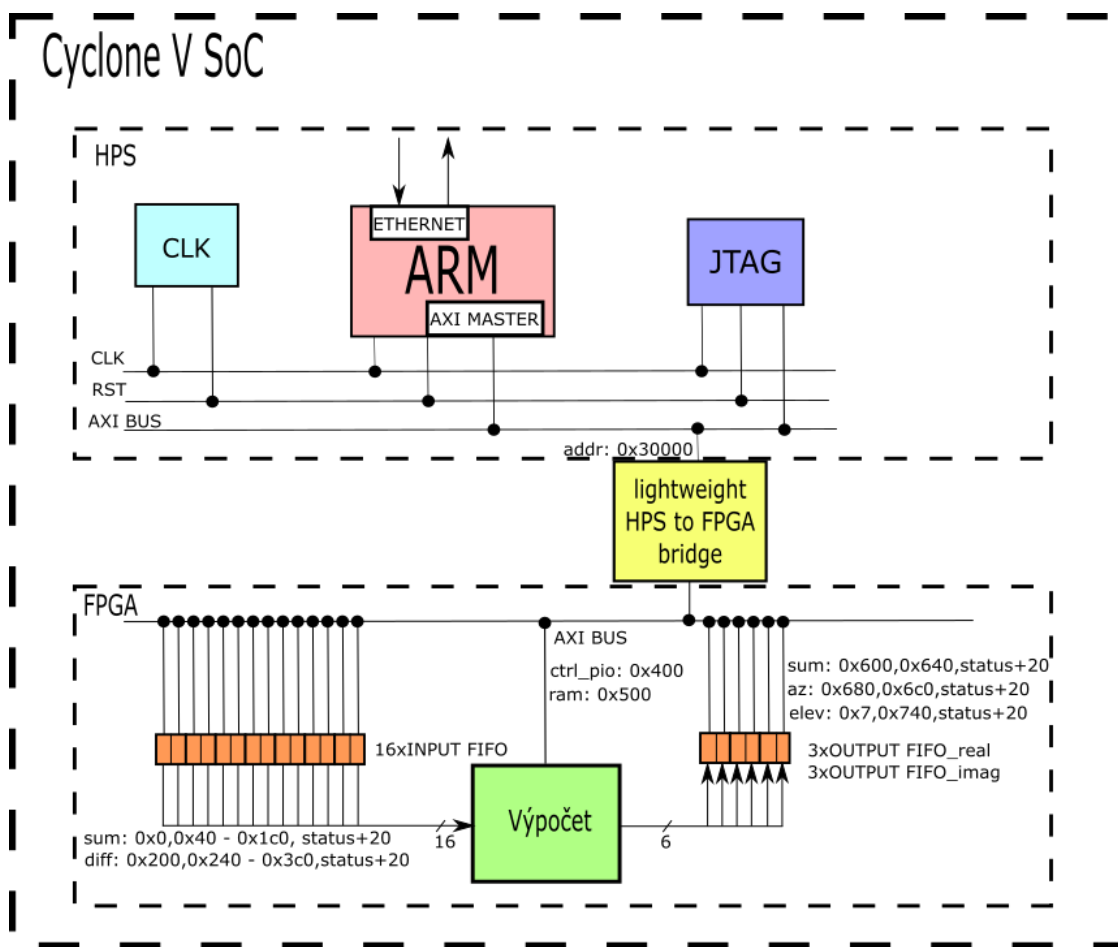
Obr. 3.5: Nastavení parametrů IP bloku paměti FIFO

3.2.3 Propojení IP Jader

Pro vytvoření top hierarchie projektu a propojení jednotlivých komponent návrhu slouží v prostředí *Intel quartus* utilita *Platform designer(Qsys)*. Pomocí grafického rozhraní je zde možné propojit jednotlivá ip jádra a zadat jejich parametry. Pro vložení do *Qsys* je nutné celý projekt přeložit v tomto prostředí a vytvořit z něj standardizované IP jádro. Nevýhodou je značné omezení podpory VHDL standardů. To znemožňuje využití některých novějších funkcí.

Po kompilaci následuje nastavení vstupních a výstupních portů v souladu se standardy Intel, aby bylo možné vzájemné propojení IP jader. Blokové schéma celého projektu je na obrázku 3.6. Obrázek přibližně odpovídá vzhledu schématu projektu v programu *Qsys*.

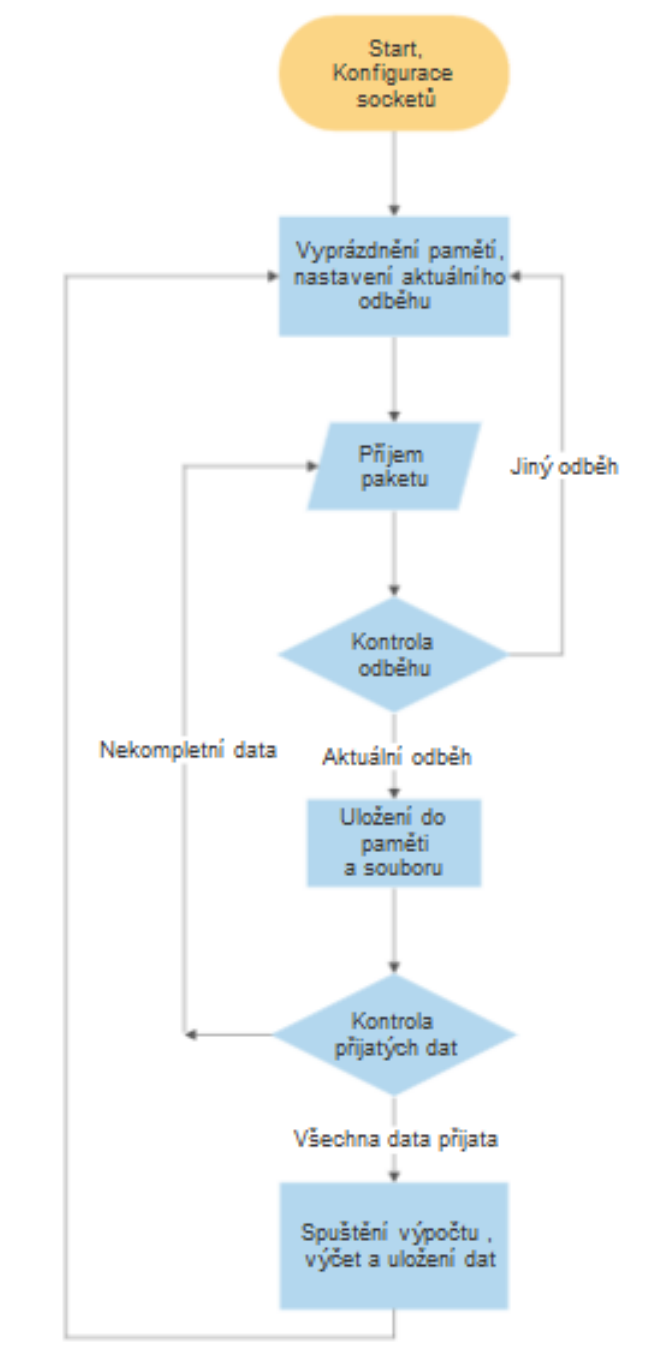
Propojení jader s procesorem je realizováno pomocí AXI sběrnice typické pro procesory ARM-Cortex A9. FPGA část je s HPS propojena pomocí zjednodušeného můstku, který je součástí architektury obvodu. Díky tomu se pro procesor jeví vstupy FPGA jako vlastní registry. Vstupy FPGA jsou tak snadno přístupné pro zápis i čtení programem běžícím v procesoru.



Obr. 3.6: Blokové schéma projektu

3.2.4 Program pro HPS

Hlavním úkolem aplikace běžící na HPS je příjem UDP paketů na 2 portech (pro součtový a rozdílový signál), jejich rozložení a uložení do paměti FIFO. Dále pak výčet výsledků a jejich odeslání. Vývojový diagram je zobrazen na obrázku 3.7.



Obr. 3.7: Vývojový diagram serverové aplikace

Pro příjem je nejprve nutné otevřít socket na portu, ze kterého je očekáván příjem. To se provádí funkcí z knihovny *sys/socket*. V případě více možných portů musí být pro každý port zvláštní socket. Tvarovač využívá 2 porty, a to 10562 pro součtový kanál a 10566 pro rozdílový kanál.

Po inicializaci socketů je nutné vyčistit paměti FIFO. Z nezjištěných příčien se totiž při zapnutí napájení do některých pamětí dostanou náhodná data.

Ta by v případě, že nebudou odstraněna, způsobila desynchronizaci dat. Paměti FIFO nedisponují možnostmi mazání nebo zrušení platnosti dat a proto je tato operace nahrazena prostým vyčítáním. I v případě nesprávných dat tak probíhá výpočet paprsků a ke smazání dojde až určením neplatnosti výstupních svazků. K mazání slouží speciální vstup výpočetního modulu, který spouští výčet a zároveň zabraňuje nahození výstupního *valid* signálu. Resetovací signál je nahozen až do doby, kdy jsou všechny vstupní paměti prázdné. Zaplnění pamětí je zjištěno přes stavové registry, které jsou rovněž přístupné z paměťového prostoru.

Samotný příjem probíhá příkazem *recvfrom*. Příkaz po spuštění čeká až do příjmu paketu. Proto při příjmu z více portů je nutné ho spouštět pro každý socket v jiném vlákně. Alternativní možností, která je využita ve tvarovači je příkaz *select*. Ten může sledovat více portů bez čekání na příjem. Čekání na příjem tak nezastaví chod programu a funkci *recvfrom* je možné volat až po zaznamenání pohybu na některém z portů. Tento příkaz pak proběhne již bez čekání.

Po přijetí jsou pakety uloženy v bytovém poli. Jejich tvar je předem definovaný, aby bylo možno jednoduše rozlišit význam dat. Přibližný tvar paketu obsahujícího operační data je v tabulce 3.1.

Tab. 3.1: Struktura přijímaného paketu [22]

Byte	Jméno	Dat.typ	Popis
1-4	Id	UInt32	Číslo odběhu
5	Row	UInt8	Číslo řady
6	Col	UInt8	Číslo sloupce
7	chan	UInt8	Kanál (rozdílový/součtový)
8	Pulse	UInt8	Pulz (dlouhý/krátký)
9-10	RSO	UInt16	Ofset dálkového segmentu
11-12	RSL	UInt16	Počet datových bytů (N)
N	QDt	Sint16	Imaginární data
N	InDt	Sint16	Reálná data

Pracovně se paket dělí na hlavičku a datovou část. Do hlavičky spadá vše od začátku až po počet datových bytů. Data jsou pouze imaginární a reálné hodnoty navzorkovaného signálu.

V hlavičce příchozích paketů je kontrolováno číslo odběhu. To je důležité především pro zachování integrity dat. Příjem dat jiného odběhu způsobí odstranění veškerých dat z pamětí a nastavení tohoto odběhu jako aktuálního. Smazáním jsou ztracena data celého odběhu. K vymazání dojde pravděpodobně pouze v případě ztráty dat některé řady.

Zároveň s ukládáním do paměti jsou data vypisována v komplexním součtovém tvaru.

Po přijetí všech 16 paketů (součtový i rozdílový kanál pro 8 řad) a předání dat do FIFO paměti je spuštěn výpočet pomocí ovládání *ready* signálu výpočetního modulu. Vyčítání skončí po příchodu posledních platných dat v paměti FIFO. Z výstupní paměti jsou následně procesorem vyčteny vypočtené paprsky. Před odesláním jsou sestaveny do výstupního paketu, obsahujícího původní hlavičku spolu s identifikací paprsku.

Pro účely ověření bylo odesílání nahrazeno ukládáním do souborů. Odeslaný UDP paket by totiž sice bylo možné zachytit, ale chybí nástroje k jeho rozebrání a porovnání s výstupem z aplikace GNU Octave.

3.2.5 Podpůrné programy

V rámci práce byla kromě samotného tvarovače vytvořena také řada podpůrných programů v prostředí Octave i v jazyce C. Jejich úkolem bylo především generování souborů nezbytných k funkci tvarovače a zajištění vzájemného přenosu dat s GNU Octave.

Generování tvarovacích koeficientů

Výpočet komplexních vah je proveden v prostředí GNU Octave. Pro výpočet byla navržena funkce *bfk*

$$[an, bn] = \text{bfk}(Lv, ap, f, N, mod, beam)$$

Do funkce se dosadí: za *Lv* rozteč prvků anténní řady,

ap odklon apertury,

f pracovní frekvence,

N počet prvků řady,

mod vektor amplitudové modulace,

beam požadovaný úhel paprsku ve stupních.

Funkci je zároveň možné využít bez zadaných parametrů, kdy budou za nezadané parametry dosazeny hodnoty pro demonstrátor 3D Fénix (viz kapitola 4). Jako výchozí funkce okna je využito Čebyševovo okno s potlačením postranních laloků 25 dB.

Výsledné komplexní matice jsou vstupem pro tvarovač. Charakter koeficientů však neumožňuje přímé použití a je nutné je nejdříve převést na vhodnější formát. Běžné hodnoty komplexních vah se pohybují v rozmezí 2 a -2 a jsou ve formátu

float. Pro práci s tímto formátem na architektuře FPGA je nezbytná hardwarová podpora, která bohužel v testovacím obvodu chybí. Řešením je převod na formát *Int16*, se kterým není problém pracovat. Před převodem se tvarovací váhy ještě vydělí s-parametry, aby tato operace nezabírala příliš mnoho zdrojů obvodu. Tuto operaci lze provést předem, protože se ani váhy ani s-parametry během provozu nemění. Poslední úpravou před převodem je násobení 10^4 , které zabrání ztrátě desetinných míst při převodu. Pro správnost výsledku je důležité v závěru tvarování toto násobení opět vykompenzovat vydělením. To je však možné až po zpětném převodu výsledku do formátu *float*.

Pro generování konfiguračního souboru paměti tvarovače byla vytvořena funkce *koef2file*. Ta bere jako vstup koeficienty vypočtené funkcí *bfk* a výsledky ukládá do textového souboru v dekadickém tvaru.

Pořadí ukládání do souboru koresponduje s připojením registrové paměti k výpočetnímu modulu a jeho dodržení je kritické pro správnost výpočtů. Seřazení koeficientů v souboru je naznačeno v tabulce.

Tab. 3.2: Struktura souboru koeficientů

Anténní řada	Kanál	Složka	Řádek
1	Součtový	Real	1
		Imag	2
	Azimut	Real	3
		Imag	4
	Elevace	Real	5
		Imag	6
2	Součtový	Real	7
		Imag	8
.	.	.	.
.	.	.	.
.	.	.	.

Data do kitu jsou následně přenesena formou textového souboru na paměťovou kartu kitu. Soubor lze kromě vyjmutí karty z kitu nahrát rovněž přes ssh.

Načtení koeficientů do paměti

Ze souboru s koeficienty na paměťové kartě jsou data nahrána do paměti RAM programem *Regs*. Program je nutné spustit ve chvíli, kdy není spuštěn server, aby nedošlo ke změně koeficientů během výpočtu.

Ukládání do paměti začíná připojením paměti do registrového prostoru procesoru. Prakticky je toto provedeno načtením adresy paměti z přípojného bodu

ve složce */dev/mem* do proměnné, která slouží jako ukazatel na začátek paměti. K počáteční adrese se dále přičítá ofset, nastavený pro danou paměť při sestavování návrhu v aplikaci Qsys. Další ofset určuje konkrétní paměťovou buňku. Adresy jednotlivých buněk jsou bytové, komunikace však probíhá vždy po 32 bitech. Paměťové buňky se tak nahrávají vždy po 4 bytech najednou. Jednotlivé koeficienty se po vyčtení z textového souboru postupně ukládají do paměti tak, že reálná a imaginární část jsou spojené do jednoho 32-bitového vektoru. Pro uložení následující hodnoty je třeba inkrementovat adresu o 4.

V případě chybného souboru je vypsána příslušná hláška oznamující typ chyby. Chybou může být špatný formát dat, které nelze převést na odpovídající číslo. Další možností je špatná velikost souboru, kdy by paměť zůstala nezaplněna nebo by se program pokoušel zapsat do paměťového prostoru mimo RAM.

Po ukončení zápisu do paměti je nutné paměť odpojit, aby bylo později možné do ní přistupovat ze serverové aplikace.

Porovnání výstupů

Po ukončení výpočtu každého svazku jsou přijatá data jednotlivých řad i výstupy tvarovače uloženy v souborech. Pro ověření správnosti je třeba tyto výsledky porovnat s modelem. Pro tyto účely byly vytvořeny další 2 skripty v prostředí Octave. Prvním z nich je funkce *bfd*, která má za úkol ze souborů dat stažených z tvarovače vyčíst data a sestavit je do matice použitelné jako vstup modelu. Výstup z modelu je opět do textového souboru se stejným tvarem jako výstup z tvarovače. Pro porovnání slouží druhá z funkcí *compare*, která odečte data obou souborů a z rozdílů hodnot vykreslí histogram odchylek.

4 Ověření funkce

Jednotlivé bloky návrhu byly ověřeny simulací v prostředí Modelsim. Výsledky však vzhledem k jejich triviální funkci sloužily pouze jako zpětná vazba při návrhu. Pro ověření správné funkce tvarovače bylo nutné srovnání vypočtených výsledků s referenčními hodnotami. Doposud nasazené DSP procesory nebylo technicky možné k tomuto účelu využít. K porovnání výstupů byl proto v rámci práce vytvořen MATLAB model v prostředí GNU Octave. Tento model byl ověřen s využitím testovacích dat se známým výstupem.

Ověření funkce samotného tvarovače bylo provedeno porovnáním výstupních dat s modelem. K porovnání sloužila reálně naměřená data radaru.

4.1 MATLAB model

Cílem modelu je realizace výpočtu součtových, azimutálních a elevačních složek svazku stejně jako u tvarovače. Model je tvořen funkcí *beamform*:

$$[B_sum, B_az, B_ele] = beamform (an, bn, sn_sum, sn_az, sn_ele, values_sum, values_dif);$$

Vstupními parametry funkce jsou: *an, bn* - komplexní váhy generované funkcí *bfk*
sn_sum - s-parametry součtového kanálu
sn_az - s-parametry azimutálního kanálu
sn_ele - s-parametry elevačního kanálu
values_sum - hodnoty signálu součtového kanálu
values_dif - hodnoty signálu rozdílového kanálu

Pro účely ověření správnosti modelu je nutné vygenerovat sadu dat s předvídatelným průběhem a výstupem. Nejvhodnějšími daty se jeví signál, který má stejnou amplitudu na všech prvcích antény, ale přichází na ni z různých úhlů v rozsahu -90 až 90°. Zbytek prostoru (za anténou) není pro ověření důležitý. V případě všesměrové antény, která neuvažuje směrovost jednotlivých prvků (což je případ modelu) by byl výstup symetrický. Výpočet testovacího signálu vychází z obrázku 1.2, kde je znázorněn dráhový rozdíl signálu dopadajícího pod úhlem.

Pro každý úhel signálu a element antény je nutné vypočítat fázové zpoždění signálu. Pomocí funkce sinus je ze vzdálenosti elementů antény a úhlu dopadu vypočítán dráhový rozdíl, který je poté vydělen vlnovou délkou (rovnice 4.1).

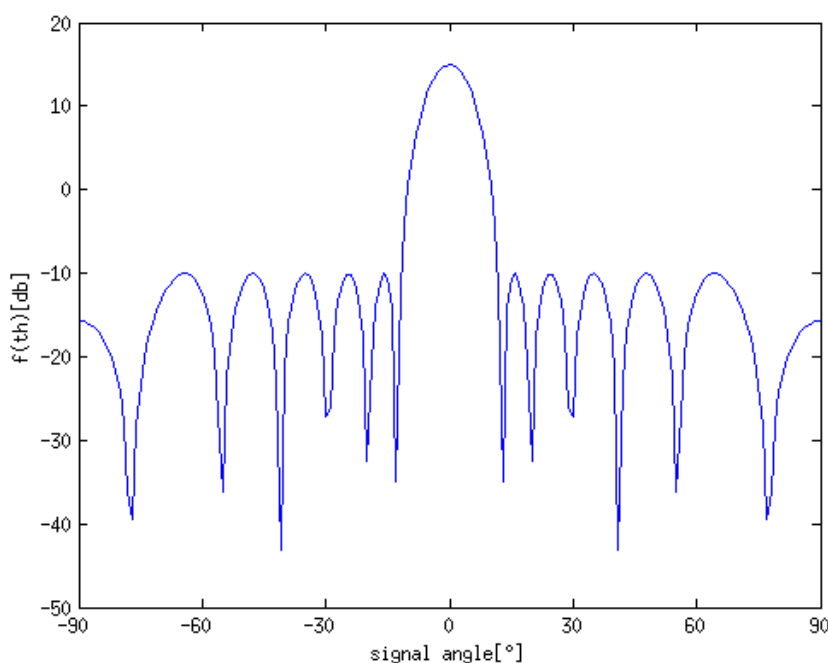
$$\Delta\phi = \frac{d \cdot \sin\theta}{\lambda} \quad (4.1)$$

Výsledné fázové zpoždění je pak aplikováno do rovnice 4.2.

$$S_n = \cos((n-1) \cdot \phi) + i \cdot \sin((n-1) \cdot \phi) \quad (4.2)$$

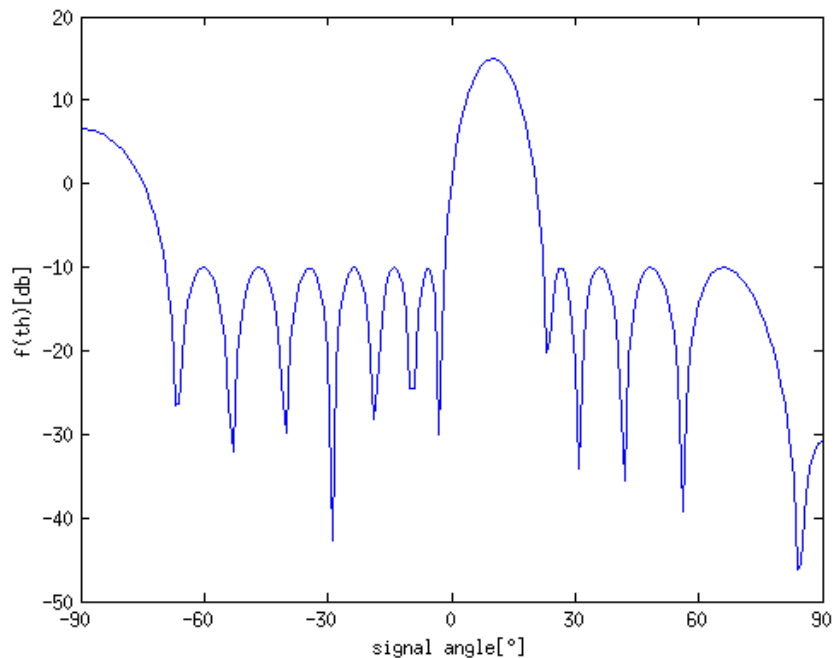
Signál prvního elementu antény tak bude roven 1 a každý další prvek bude posunutý o fázový rozdíl oproti předchozímu.

Testovací signál by po průchodu tvarovačem měl v součtovém kanálu vykazovat nejvyšší hodnoty zisku v místě, kam je namířen paprsek antény. Rozdílový signál bude naopak v ose antény mít nejmenší zisk. Na obrázku 4.1 je vykreslen přijatý součtový signál při nulovém vychýlení svazku.



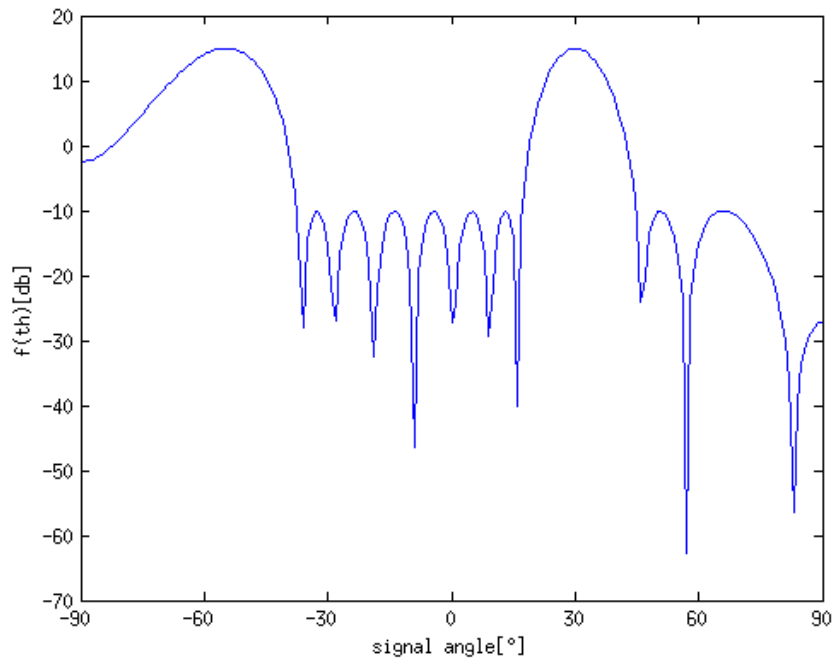
Obr. 4.1: Součtový signál svazku při nulovém vychýlení od osy antény

Největší zisk je dle očekávání v ose antény. Postranní laloky mají stejnou velikost, která je dána parametrem Čebyševovy funkce použité v komplexních vahách. V tomto případě jsou postranní laloky potlačeny o 25 dB. Pokud se bude vychýlení zvětšovat, bude se zároveň s tím posouvat i oblast nejvyššího zisku (obrázek 4.2).



Obr. 4.2: Difrakční lalok při vychýlení 10°

Použitá anténa má při dané frekvenci vysílání poměr vzdálenosti elementů a vlnové délky větší než 0,5. Při určité velikosti vychýlení, které překračuje parametry antény se tak objeví difrakční laloky (obrázek 4.3). Ty mají stejnou velikost jako hlavní lalok [4].



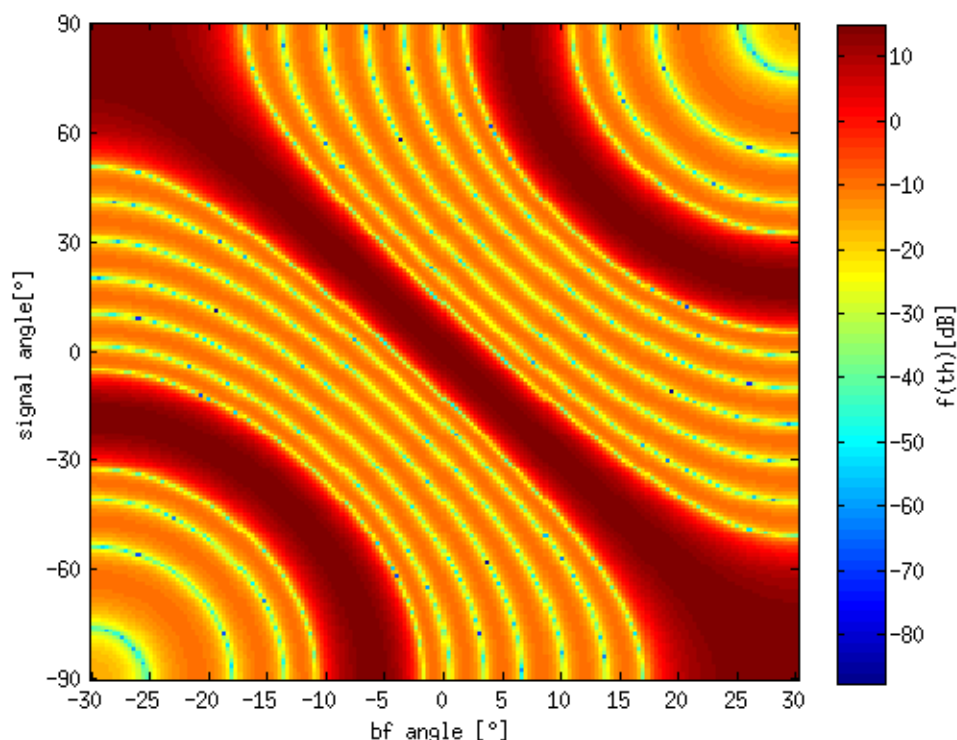
Obr. 4.3: Difrakční lalok při vychýlení 30°

Fyzické parametry antény jsou pevně dané a jediná možnost jak se zbavit difrakčních laloků je omezení maximální velikosti vychýlení. Omezení lze pozorovat z obrázku 4.4, kde je vykreslen výstup součtového signálu pro všechny úhly vychýlení.

Úhly, pro které vznikají difrakční laloky (v rozích grafu), nelze využít k vychylování, aby nemohlo dojít k záměně difrakčního a hlavního laloku. Maximální vychýlení svazků je dáno rovnicí 4.3.

$$\frac{\Delta L_v}{\lambda} = \frac{1}{1 + \sin(\theta_{max})} \quad (4.3)$$

Po dosazení parametrů demonstrátoru vychází poměr vzdálenosti prvků antény ku vlnové délce 0,76, což vede k maximálnímu úhlu vychýlení 18°. Při fyzickém natočení antény o 15° od země je tedy možné paprsek vychylovat v rozmezí 33 a -3°.



Obr. 4.4: Průběh součtového signálu při vychylování svazku

4.2 Porovnání tvarovače s modelem

Testovací data pro tvarovač pochází z demonstrátoru 3D radaru Fénix [22], který je umístěný na střeše sídla firmy Retia a.s. v Pardubicích (obrázek 4.5).

Zařízení je vybaveno fázovanou anténou složenou z 8 vertikálních řad, které tvoří 32 prvků. Rozteč jednotlivých elementů je 73,1 mm a odklon od vodorovné osy 15°. Radar využívá tvarování svazků pouze pro elevaci. Azimut je měřen z polohy rotující antény. Zařízení pracuje jako aktivní primární radar v pásmu S na frekvenci 3120 MHz.

Využitá data byla za provozu radaru vytvořena záznamem datových toků programem Wireshark. Pro simulaci reálného provozu byl celý návrh nahraný ve vývojovém kitu propojen pomocí sběrnice Ethernet s počítačem, na kterém byl přehráván naměřený záznam.

Při přehrávání záznamu se vyskytl problém ve složení UDP paketů v záznamu. Při původním provozu, kdy byl záznam pořízen, proběhla mezi jednotlivými prvky v síti ARP komunikace a následné přenosy byly řízeny fyzickými adresami bez ohledu na IP adresu. Při použití jiného než původního tvarovače nebyl možný příjem dat. Řešením tohoto problému může být změna MAC adresy kitu na adresu původního

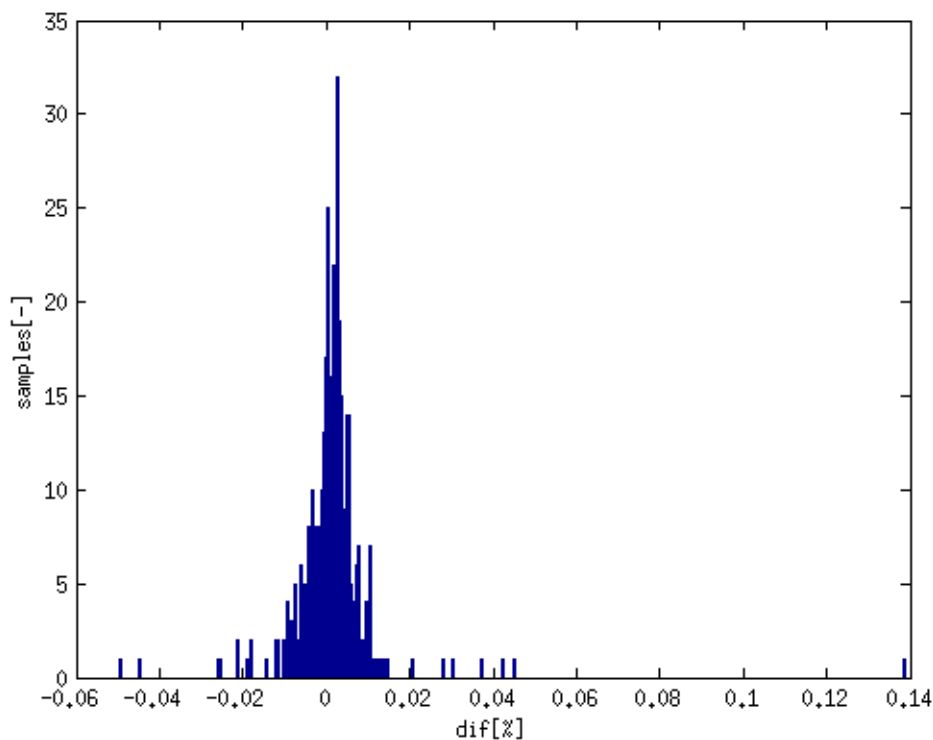


Obr. 4.5: Demonstrátor 3D radaru na střeše firmy retia a.s.

signálového procesoru, nebo jak tomu bylo i v tomto případě, přepsání cílové adresy u všech paketů v záznamu na adresu kitu. Přepis byl proveden funkcí *tcprewrite*, která je součástí programu *tcpreplay*. Po přepsání bylo již možné pakety přijímat.

Výsledky z tvarovače byly uloženy do textového souboru čitelného podpůrným skriptem v Octave. Ten byl později pomocí ssh připojení nakopírován z paměťové karty kitu do počítače pro porovnání s modelem. Tvarovač zajistil kromě textového výstupu výsledků i převod přijatých dat z paketů do formy zpracovatelné modelem.

Data radaru byla převážně tvořena šumem a korektnost obrazového výstupu by proto nebylo možné ověřit. Výsledky obou výpočtů tak byly pouze porovnány a odchylky v procentech vyneseny do grafu 4.6.



Obr. 4.6: Histogram odchylek vzniklých zaokrouhlováním

4.3 Vyhodnocení

Funkce tvarovače byla testována porovnáním s referenčním modelem vytvořeným v GNU Octave. Funkce navrženého obvodu byla ověřena. Vzniklé odchylky od modelu byly zapříčiněny zaokrouhlováním při převodu mezi formáty *float* a *Int16*. Jejich velikost je ve většině případů menší než 0,02 %, což je pro praktické aplikace zanedbatelná hodnota.

4.4 Využití zdrojů obvodu FPGA

Požadavkem na návrh tvarovače bylo mimo jiné stanovení maximálního počtu realizovatelných svazků na jedné desce. Počet svazků závisí na složitosti návrhu a počtu dostupných zdrojů daného obvodu. Z tabulky 4.1 je patrné využití zdrojů obvodu 5CSEMA4U23C6N. Využití je stanoveno při 16-bitových vstupech i koeficientech.

Tab. 4.1: Využití zdrojů obvodu

Prostředek	Dostupný počet	Tvarovač s 1 svazkem	Každý další svazek
ALM	15880	7891	2133
DSP	84	48	48
HW násobička	168	0	0
Registr	60376	16138	3776
Paměť	2700kb	626kb	0
RAM blok	270	125	0
ARM jádro	2	1	0

Pro přidávání dalších svazků je kritický počet HW násobiček. Využití celého tvarovače a samotného svazku bylo simulováno nezávisle. Může se tak jevit, že je využito více DSP bloků než je v obvodu dostupných. Při současném zapojení více svazků je však funkce chybějících DSP bloků nahrazena nezávislými násobičkami. Při maximálním využití daný obvod umožňuje počítání až 3 svazků paralelně. Použití plánovaného obvodu z rodiny Aria 10 umožní při současném rozlišení výpočet 7-60 paprsků dle konkrétního obvodu.

Maximální pracovní frekvence byly stanoveny statickou časovou analýzou při napájecím napětí 1,1V a teplotách 0 a 85°C

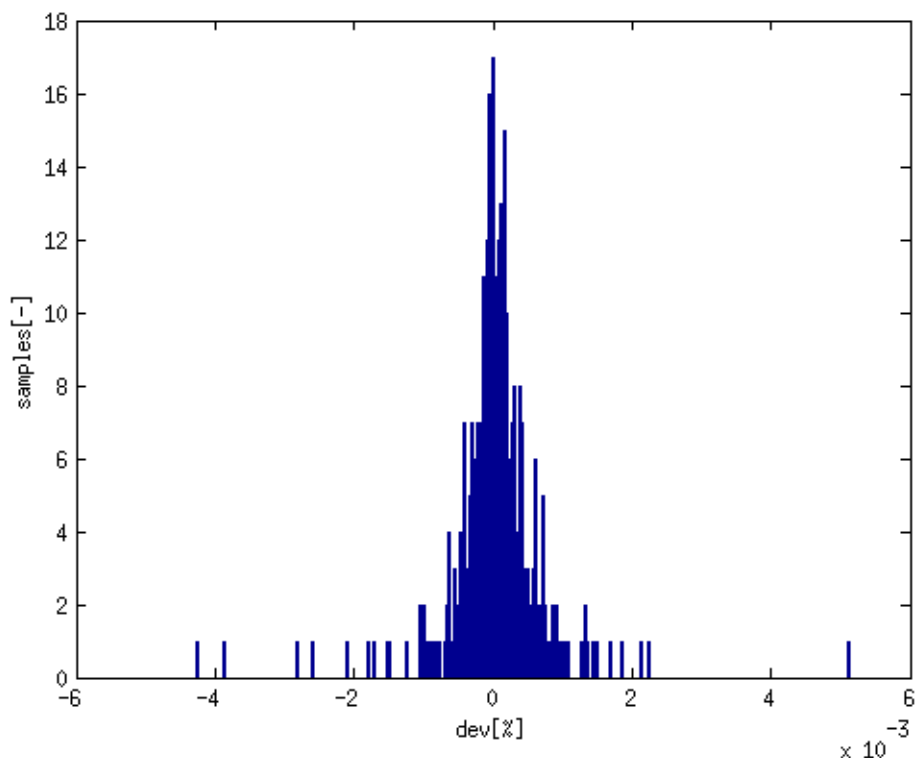
Tab. 4.2: Statická časová analýza

Teplota	FPGA	AXI	ARM
85°C	73,35 MHz	99,14 MHz	717,3 MHz
0°C	78,52 MHz	99,14 MHz	1251,56 MHz

4.5 Zvyšování přesnosti

Pro většinu aplikací je chyba zaokrouhlování zanedbatelná, ale v případě nutnosti je možné další zmenšení chyby. K tomu je možné využít 2 metod.

První možnost je dána fyzickou stavbou obvodu. Hardwarové sčítačky obsažené v architektuře totiž disponují šířkou 18x18 bitů. Prosté rozšíření koeficientů by znamenalo zvýšení přesnosti vstupů o 1 desetinné místo. Praktický dopad této změny byl simulován a výsledkem bylo zlepšení o 1 řád při zachování téměř stejného využití obvodu. Výsledek je znázorněn na obrázku 4.7.



Obr. 4.7: Odchyly výsledků po rozšíření koeficientů

Pro praktickou realizaci tohoto rozšíření by však bylo nutné přidat další blok zajišťující bitové přizpůsobení pro komunikaci s HPS. Systém HPS respektive odlehčený můstek mezi HPS a FPGA je totiž schopen pracovat pouze se šířkou 32 bitů a tomu je přizpůsobena i velikost paměti, která je složena ze 32-bitových bloků.

Druhou možností je pracovat přímo s formátem float. Tato možnost je z hlediska přesnosti nejlepší, ale zároveň nejméně hospodárná na využití obvodu. V případě Cyclone V není tato možnost přípustná vzhledem k absenci HW podpory pro float. Při využití obvodu Aria 10 již float podporován je, nicméně se značně sníží maximální počet tvarovaných svazků. Samostatné HW násobičky není možné pro float využít a na jedno násobení bude využit celý DSP blok, který dříve umožňoval 2 násobení. I v případě využití nejvýkonnějších obvodů dané řady tak bude možné dosažení pouze 15 svazků.

4.6 Další rozšíření

Důvodem návrhu tvarovače v této práci bylo ověření principu tvarování a možnosti jeho implementace do obvodu FPGA. Pro praktické využití v radaru je nezbytné doplnění některých funkcí, které nebyly v této práci uvažovány.

Téměř nezbytným rozšířením je kalibrace tvarovacích koeficientů za běhu, kdy se koeficienty pro každý odběh násobí hodnotami přijatými v kalibračním paketu radaru.

Dalším vhodným doplněním je využití potlačení postranních laloků (SLC) [14], které zabraňuje rušení radaru signály z jiných zdrojů.

Závěr

V rámci práce byl proveden návrh, implementace a ověření funkce algoritmu tvarování příjmových anténních svazků 3D radaru.

Úvodní část práce obsahuje témata nezbytná k odvození algoritmu tvarování a jeho využití. Z uvedených informací by měl být zřejmý význam veškerých použitých proměnných a funkcí stejně jako důvod jejich volby. Zároveň je zde probráno zařazení tvarovače do struktury radaru.

Praktická část řeší popis implementace tvarovače do vývojového kitu s obvodem Intel Cyclone V SoC. Detailně jsou zde popsány jednotlivé bloky návrhu, jejich propojení a řídicí software. Pro tvarovač byla v rámci práce rovněž vytvořena řada podpůrných programů.

Implementovaný algoritmus tvarování je konfigurovatelný a je možné ho využít v libovolném zařízení. Části návrhu zajišťující komunikaci s okolím jsou vázány na použitý obvod a pro přenos bude nutná jejich změna. Konfiguraci obvodu nelze provádět za provozu, ke změně parametrů je nutné opakovat překlad.

Funkce tvarovače byla po nahrání do vývojového kitu porovnána s referenčním MATLAB modelem s pozitivním výsledkem. Vytvoření tohoto modelu a ověření jeho správnosti bylo rovněž součástí práce.

Z modelu také vyplývá, že při konfiguraci radaru, která byla využita pro ověření funkce tvarovače, bude kvůli existenci difrakčních laloků možné vychylovat pouze v omezeném rozsahu úhlů. Tento nedostatek nelze bez změny fyzické konfigurace antény odstranit.

Literatura

- [1] MAILLOUX, Robert J. *Phased array antenna handbook*. 2nd ed., Boston: Artech House, 2005. ISBN 978-1-58053-689-9.
- [2] L.S. Benenson, V.A. Zhuravlev, S.V. Popov, G.A. Postov, *Antenna arrays*, Ed. L.S. Benenson. Moscow: Sov. Radio, 1966. – 368 P.
- [3] MAHAFFZA, Bassem R. *Introduction to radar analysis*. Boca Raton: CRC Press, c1998. ISBN 0849318793.
- [4] HANSEN, Robert C. *Phased array antennas*. New York: Wiley, c1998. ISBN 047153076X.
- [5] Wolff, Ch.:*Radartutorial*,[Online],Lechfeld 1998[cit. 14.05.2019], URL:<http://www.radartutorial.eu/index.html#this>
- [6] Moernaut, G.J.K, Orban, D. *The Basics of Antenna Arrays*,Orban Microwave Products,[Online][cit.14.05.2019], URL:<https://orbanmicrowave.com/the-basics-of-antenna-arrays/>
- [7] Kumar, G. *Antena arrays course*, [Online], IIT Bombai,[cit. 14.05.2019], URL:<https://nptel.ac.in/courses/108101092/Week-4-Antenna-Arrays-II.pdf>
- [8] LI, Jian a Petre STOICA. *Robust adaptive beamforming / edited by Jian Li and Petre Stoica*. Hoboken, NJ: John Wiley, 2006. ISBN 0471678503.
- [9] BALANIS, Constantine A. *Antenna theory: analysis and design*. 3rd ed. Hoboken: Wiley-Interscience, 2005. ISBN 978-0-471-66782-7.
- [10] Shejbal, T.*Aktivní fázované anténní systémy pro přibližovací radary*, Pardubice 2016, Disertační práce, FEI UPCE, Školitel P.Bezoušek
- [11] Radar Handbook ; editor in Chief M. I. Skolnik. 2. ed. New York: McGraw-Hill Publishing Company, 1990.
- [12] SHERMAN, Samuel M. a David Knox BARTON. *Monopulse principles and techniques*. 2nd ed. Boston: Artech House, c2011. ISBN 9781608071746.
- [13] O'Donnell, R.M, *Radar system Engineering course*, [Online přednášky], IEEE New Hampshire 2010, [cit.14.05.2019], URL:<http://aess.cs.unh.edu/Radar%202010%20PDFs/>

- [14] BEZOUŠEK, P. ŠEDIVÝ, P. *Radarová technika*. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-03036-9.
- [15] Mathworks, *Taylor window function*, [Online manuál], [cit 14.05.2019], URL: <https://www.mathworks.com/help/signal/ref/taylorwin.htm>
- [16] Mathworks, *Chebyshev window function*, [Online manuál], [cit 14.05.2019], URL: <https://www.mathworks.com/help/signal/ref/chebwin.htm>
- [17] Šedivý, P.: *Rádiové systémy*, Školení Retia a.s., 2018 - interní dokument.
- [18] Cetinoneri, Berke, A. Atesal, Yusuf, M. Rebeiz, Gabriel. (2011). *An 8 × 8 Butler Matrix in 0.13-μm CMOS for 5–6-GHz Multibeam Applications*. Microwave Theory and Techniques, IEEE Transactions on. 59. 295 - 301. 10.1109/TMTT.2010.2097751.
- [19] Procházka, T., Bartoň, Z. *s-parametry*, [Online], Elektrevue 29/2002, [cit.14.05.2019], URL: <http://www.elektrevue.cz/clanky/02029/index.html>
- [20] TeasIC, *DE0-nano-SoC/Atlas-SoC Kit*, [Online katalogový list], 2015, [cit.14.05.2019], URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=163&No=941&PartNo=4>
- [21] Intel, *Cyclone V Device Overview*, [Online katalogový list], [cit.14.05.2019], URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_51001.pdf
- [22] Pár, D.: Popis signálového zpracování Re3D, Retia a.s. - interní dokument.
- [23] Intel, *On-Chip FIFO Memory Core*, [Online katalogový list], [cit 14.05.2019], URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/nios2/qts_qii55002.pdf
- [24] Intel, *Embedded Peripherals IP User Guide*, [Online katalogový list], [cit 14.05.2019], URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_embedded_ip.pdf
- [25] Intel, *Avalon Interface Specifications*, [Online katalogový list], [cit.14.05.2019], URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/mnl_avalon_spec.pdf

Seznam symbolů, veličin a zkratek

FPGA	Field programmable gate array - Programovatelné hradlové pole
DSP	Digital Signal Processing –Číslicové zpracování signálů
IEEE	Institute of Electrical and Electronics Engineers – Institut pro elektrotechnické a elektronické inženýrství
AESA	Active electronically scanned array – Aktivní elektronicky řízená fázovaná řada
PESA	Passive electronically scanned array – Pasivní elektronicky řízená fázovaná řada
Soc	System on Chip – Systém na čipu
HPS	Hard Processor system – Integrovaný procesor v systému
ARM	Advanced RISC Machine – Procesor s jádrem firmy ARM
GNU	GNU's Not Unix – Svobodný software inspirovaný unixem
RAM	Random access memory – Paměť s náhodným přístupem
UDP	User Datagram Protocol – Internetový protokol bez záruky doručení
DTR	Digital Transceiver Receiver – Digitální vysílací a příjmový modul
FIFO	First in first out – Paměťová fronta
IP	Intellectual Property – Licencované jádro/funkční blok
VHDL	VHSIC Hardware Description Language – Jazyk pro popis hardware
HW	Hardware – Pevně vytvořená část obvodu
MAC	Media Access Control – Fyzická adresa síťového zařízení
A/D	Analog Digital Converter – Převodník spojitého signálu na číslicový
SP	Short pulse – Krátký pulz
LP	Long pulse – Dlouhý pulz
Qsys	Altera Platform designer – Software pro tvorbu top hierarchie projektu